2020

Assessing Robustness of the Rasch Mixture Model to Detect Differential Item Functioning - A Monte Carlo Simulation Study

Jinjin Huang

Assessing Robustness of the Rasch Mixture Model to Detect Differential Item Functioning - A Monte Carlo Simulation Study

---

A Dissertation

Presented to

the Faculty of the Morgridge College of Education

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

---

by

Jinjin Huang

December 2020

Advisor: Kathy E. Green, Ph.D.

Author: Jinjin Huang
Title: Assessing Robustness of the Rasch Mixture Model to Detect Differential Item Functioning - A Monte Carlo Simulation Study
Advisor: Kathy E. Green, Ph.D.
Degree Date: December 2020

Abstract

Measurement invariance is crucial for an effective and valid measure of a construct. Invariance holds when the latent trait varies consistently across subgroups; in other words, the mean differences among subgroups are only due to true latent ability differences. Differential item functioning (DIF) occurs when measurement invariance is violated. There are two kinds of traditional tools for DIF detection: non-parametric methods and parametric methods. Mantel Haenszel (MH), SIBTEST, and standardization are examples of non-parametric DIF detection methods. The majority of parametric DIF detection methods are item response theory (IRT) based. Both non-parametric methods and parametric methods compare differences among subgroups categorized by observed covariates such as gender and grade. As a result, the differences within unobserved subgroups are likely to be neglected. The Rasch mixture model (RMM), a combination of the Rasch model and mixture model, is an alternative for extracting the latent class (LC) from summarizing similar identities of underlying latent traits. DIF can be calculated among LCs based on the differences among mean item difficulties for each LC.

The purpose of this study was to examine the robustness of the RMM in detecting DIF through manipulating five variables: number of items (i.e., test length, 2 levels), proportion of DIF (3 levels), LC structure (2 levels), group size (2 levels) and DIF type (2 levels), which yields 2*3*2*2*2 = 48 scenarios. A sample size of 3,000 was used for each replication of each scenario. The robustness of the RMM on detecting DIF was

assessed from two perspectives: latent class structure recovery and parameter recovery. One hundred replications per scenario were used for LC structure recovery and 200 replications per scenario were used for parameter recovery.

The main and interactions effects of five manipulated factors on LC structure recovery and parameter recovery were examined by conducting factorial analysis of variance (ANOVA). Both AIC and BIC showed a conservative pattern on LC structure recovery in which the recovered LCs did not match the true structure perfectly or even in the majority of cases. That is, it was rare that the correct latent structure was recovered at 100%. For classifier recovery, all five manipulated factors showed effect sizes that were medium or larger except DIF type ($\eta^2 < 0.06$), and there were two medium effect size interactions for classifier recovery, and they were number of items by group size interaction ($\eta^2 = 0.10$) and LC structure by group size interaction ($\eta^2 = 0.13$). There were three main and three interaction effects of the five manipulated factors on DIF recovery ($\eta^2 > 0.06$) and they were effects of number of items, proportion of DIF items, LC structure, number of items by LC structure interaction, proportion of DIF items by LC structure interaction, and DIF type by LC structure interaction. Among these effects, group size ($\eta^2 = 0.45$) had the strongest effect on classifier recovery and LC structure ($\eta^2 = 0.86$) had the strongest effect on DIF recovery. It is recommended for practitioners to have close group sizes for latent classes, 20% to 40% proportion of DIF items, and a LC structure close to a two LC structure, to determine DIF using an RMM. Both AIC and BIC are not suggested as model selection methods in DIF detection using the RMM. Instead the Cressie-Read statistic can be an option for choosing the correct number of

latent classes from observed response patterns as the Cressie-Read statistic includes statistical tests rather than using likelihood ratios. A practitioner can identify DIF and its direction through calculating the item difficulty difference Δb between two latent classes. It can be considered as no item DIF for using the RMM method when $\Delta b < 0.3$, small DIF when $0.3 \leq \Delta b < 0.9$, medium DIF when $0.9 \leq \Delta b < 1.5$, and large DIF when $\Delta b \geq 1.5$.

Finding more reliable model selection indices for the RMM on DIF detection, increasing the efficiency of simulation, and including a single latent class structure as a comparison are directions for future study. The number of replications used in this study is recommended for practitioners who want to conduct simulation studies using the Rasch mixture model.

Acknowledgements

I would like to thank Dr. Kathy Green for her patience and continuous supports through my PhD years. Without her, all these works cannot be done. I would also like to thank my father Dr. Guoyu Huang for his unconditional supports on me. He is a good father and a great role model for me.

Table of Contents

List of Tables

List of Figures

Chapter One: Introduction and Literature Review

Introduction

In social science measurement, invariance or test equivalence is crucial when designing scales to measure constructs. Standardized tests and questionnaires support decision making across multiple disciplines such as education, business, and medicine. But, in order to be useful for making decisions regarding population subgroups, tests and questionnaires must consistently reflect the construct for each of those subgroups. Measurement invariance is gained when the latent trait underlying a scale varies consistently with observed scores across subgroups. If invariance holds at the item level, the mean differences across subgroups are due to the true difference in participants' ability to endorse an item, which is usually called the theta (θ) value. In other words, invariance holds if the item is functioning similarly for different subgroups. If not, differential item functioning (DIF) occurs, which contaminates the validity and reliability of a test and calls assessment of group differences into question. At a more fundamental level, the existence of DIF jeopardizes fairness in testing. The present study examined identification of DIF under diverse factors using one item response theory model, the Rasch mixture model.

Differential functioning at the test level is usually called differential test functioning (DTF). It is the combined effect of DIF on all of the items that reflect a certain construct. Each individual DIF item may exhibit a different direction of effect on

a test, which may be cancelled out and thus reach an acceptable level of DTF, but it makes DTF a questionable metric of measurement invariance.

It is likely that subgroups of people who are from different cultural backgrounds and had different educational opportunities would obtain different mean scores on a performance test. However, differences such as knowledge, skills, and developed abilities may not be the only source of score differences. Score differences may also be due to artificial differences like the testing process or biased items. Maintaining measurement invariance is an effort to distinguish true differences on latent variables from differences across different subgroups caused by the testing process (Green et al., 1989). Measurement invariance can be regarded as the most important preliminary foundation of an effective measure.

As invariance is crucial, considerable attention has been devoted to its appraisal under both classical test theory and item response theory. A summary of that literature follows. While many methods of DIF identification and statistical tests for DIF magnitude have been proposed, one of the more recent approaches is use of the Rasch mixture model. It is problematic to treat groups based on manifest variables such as gender and ethnicity as homogenous (Samuelsen, 2005). The Rasch mixture model is able to identify the distribution for each manifest group or covariate within latent classes (Samuelsen, 2005). Preinerstorfer and Formann (2012) examined parameter recovery of the Rasch mixture model for different test lengths and sample sizes when there were two subgroups of equal size; parameter recovery worked well in their study. Frick et al (2015) investigated the influence of different magnitudes of DIF and different ability

distributions applying the Rasch mixture model in DIF detection. However, the influence of different proportions of DIF items in a test and an uneven size of subgroups have not been explored. This study discusses the framework of the Rasch mixture model and elements of its usage for DIF detection. A Monte Carlo simulation was proposed to investigate the influence of test length, number of latent classes, the proportion of DIF items, and magnitude of DIF on parameter recovery in the Rasch mixture model.

Overview of DIF Methods

There are two types of DIF: uniform DIF and nonuniform DIF. Uniform DIF means that an item is equally probable to be endorsed by one group compared to another group (Sternberg & Thissen, 2006). That is, the DIF is uniform for members of a group compared to a reference group across all levels of the trait continuum so the group has a consistent advantage/disadvantage on an item. Uniform DIF is independent of the common ability level of a group. When the DIF is associated with participants' ability or tendency to endorse an item (θ), it is called nonuniform DIF. With nonuniform DIF, the advantage/disadvantage of a person in a group depends on where on the trait continuum the person's trait score falls. At the lower end of the trait continuum, the person in the group may be advantaged but at the upper end of the continuum, the person may be disadvantaged. Although non-IRT and IRT methods have different forms of detecting non-uniform DIF, they ask same question: it the association between manifest groups and item functioning homogeneous across all test takers' scores.

There are many methods and criteria for measuring, detecting, and flagging the magnitude of DIF. These methods either measure the magnitude of DIF or provide

statistical tests for group differences in DIF based on asymptotic properties of distributions of effect sizes or parameters. Fundamentally, DIF methods can be categorized into two camps: ones with an item response theory (IRT)-based approach and ones with a non-IRT-based approach. Non-IRT methods are usually called traditional methods for detecting DIF and most of them are for detecting uniform DIF. Mantel Haenszel (MH), SIBTEST, and standardization are similar DIF detection methods, and they are all based on contingency table statistics (Magis et al., 2010). Several researchers have modified MH (Mazor et al., 1994) and SIBTEST (Li & Stout, 1996; Finch & French, 2007) for detecting non-uniform DIF, which can be referred as non-uniform Mantel Haenszel (NU-MH) and non-uniform SIBTEST (NU-SIBTEST). The logistic regression method (Swaminathan & Rogers, 1990) sits between non-IRT and IRT methods. It models the logit of the probability of a person endorsing a certain item as the dependent variable from a linear combination of several independent variables such as group classifier, total test score, and the interaction between group classifier and total score. Logistic regression methods can be used to detect both uniform and non-uniform DIF. The following section provides an introduction to IRT as a preface to understanding how DIF is assessed.

Item Response Theory

Comprising a class of models and methods, IRT allows comprehensive analysis of responses at the item level of a test or construct (Steinberg & Thissen, 2006). IRT models assume that the probability of a correct or desired response is a mathematical function of person and item parameters. Often referred to as latent trait models, IRT models capture

hypothesized traits, attributes, and constructs which cannot be directly observed from inferring manifest discrete responses. Arguably the most often used is unidimensional IRT which assumes that observed item responses depend on a single continuous construct (Steinberg & Thissen, 2006). IRT requires local independence of items which means (1), the probability of the correct or desired option of an item is not related to the rest of items on the scale, and (2) responses to each item is each person's independent decision.

For a dichotomous item, a 3-parameter (3PL) IRT model can be written as:

$$P(X_{ij} = x_{ij}|\theta_i, a_j, b_j, c_j) = c_j + (1 - c_j)\frac{e^{a_j(\theta_i - b_j)}}{1+e^{a_j(\theta_i - b_j)}} \quad (1)$$

where x_{ij} is response of subject i to item j, θ_i is the ability or tendency of subject i; a_j, b_j and c_j are parameters for discrimination, difficulty and psudo-guessing of item j respectively. A 2PL IRT model can be obtained from Equation 1 by fixing c_j to 0; and by also fixing a_j to 1 the 1PL model is obtained. The one-parameter model can be written as:

$$P(X_{ij} = x_{ij}|\theta_i, b_j) = \frac{e^{(\theta_i - b_j)}}{1+e^{(\theta_i - b_j)}} \quad (2)$$

Although the Rasch model shares same mathematical form with 1PL IRT model as showed in equation (2), it takes an entirely different perspective of conceptualizing the relationship between data and theory. Unlike IRT, which emphasizes the primacy of fitting a model to observed data, the Rasch model emphasizes superiority of the model; thus, misfitting items and persons are eliminated as nonresponsive to the task. The differences between an IRT and Rasch paradigm may be puzzling for new researchers reading literature and applying IRT in new fields (Andrich, 2004). The term "specific objectivity" was used by Rasch (1966; Perline et al., 1979) to describe the characteristic

of his model: (1) comparison of two subjects/persons is independent of which instruments are used to measure them, (2) comparison of two instruments is independent of the subjects on whom they are used. The "specific objectivity" accounts for the Rasch model's most important property of invariance (Andrich, 2004).

IRT Methods for Uniform DIF Detection

IRT methods can be used for detecting DIF. Specifically, 1PL only has the capacity of detecting uniform DIF but a 2PL and 3PL can be used to detect both uniform and non-uniform DIF. It is assumed that if measurement invariance holds across different subgroups, then the parameters of IRT models (e.g., item logit position) for subgroups should be statistically equivalent; that is, parameters are the same within sampling error. Steinberg and Thissen (2006) provide guidelines for reporting DIF effect size based on parameter differences across subgroups for both dichotomous and polytomous IRT models. One can compare DIF among subgroups by directly subtracting parameters for selected subgroups, though they did not cite any statistical significance test of the parameter difference (Steinberg & Thissen, 2006). There are three major IRT-based DIF detection methods and they are the likelihood ratio test (LRT—Thissen et al., 1988), Lord's chi-square (Lord, 1980), and Raju's area measure (Raju, 1988).

Instead of directly comparing the magnitude of DIF in IRT parameters, the LRT compares the likelihood of a compact model (in which parameters are constrained to be the same across different subgroups) to an augmented model (in which parameters of interest are free to vary for different subgroups). The resulting statistics approximate a chi-square distribution with degrees of freedom equal to the difference in the number of

parameters estimated in the augmented model compared to that in compact model. If the test statistic is statistically significant, then DIF occurs among the parameters of interest in the augmented model. LRT is defined as:

$$G^2 = -2\ln\frac{L(Model_{compact})}{L(Model_{augumented})} \sim \chi^2 \quad (3)$$

Rather than using a fully constrained-baseline model in traditional LRT, Stark et al. (2006) suggest using a free-baseline model in which all parameters are free to vary across different subgroups. By constraining parameters of interest one at a time for an augmented model, the significance of G^2 is tested using Bonferroni corrected critical p values. They conclude that a free-baseline model is more effective than a constrained-baseline model.

The main idea of Lord's chi-square test (Lord, 1980) is to equate a vector of IRT parameters of focal groups to a vector of IRT parameters of a reference group as the null hypothesis. A premise of Lord's chi-square test is that a common metric must be used to scale all tested item parameters (Candell & Drasgow, 1988). Lord's chi-square method has the following form:

$$Q_j = (v_{jR} - V_{jF})'(\sum_{jR} - \sum_{jF})^{-1}(v_{jR} - v_{fR}) \quad (4)$$

where $v_{jR} = (a_{jR}, b_{jR}, c_{jR})$ and $v_{jF} = (a_{jF}, b_{jF}, c_{jF})$ are, respectively, vectors of the jth item's discrimination, difficulty, and pseudo-guessing of the reference group and the focal group. $\sum_{jR}$ and $\sum_{jF}$ are the variance-covariance matrices for corresponding parameters. The Q_j has an asymptotic chi-square distribution with degrees of freedom equal to the number of tested parameters in the model. For the Rasch model, Equation 3 can be written as:

$$Q_i = \frac{(b_{jR} - b_{jF})^2}{\hat{\sigma}_{jR}^2 + \hat{\sigma}_{jF}^2} \quad (5)$$

where $\hat{\sigma}_{jR}$ and $\hat{\sigma}_{jF}$ are, respectively, standard errors for the reference and focal group's jth item difficulty parameters.

Raju's area measure (Raju, 1988) computes the signed area between item characteristic curves of the focal and reference groups. If DIF does not exist, the signed area should be zero, which accounts for the null hypothesis of Raju's method. Similar to Lord's chi-square test, parameters of interest should be on a common metric. A crucial restriction for Raju's area measure is that pseudo-guessing parameters for compared subgroups should be set equal (Raju, 1988). Specifically, for the Rasch model, the test statistic of Raju's method is identical to the square root of that for Lord's chi-square, given in equation 4:

$$Z = \frac{b_{jR} - b_{jF}}{\sqrt{\hat{\sigma}_{jR}^2 + \hat{\sigma}_{jF}^2}} \quad (6)$$

These three IRT-based DIF methods only have test items and a grouping variable which is dichotomous as independent variables. However, reality is much more complicated and usually DIF may occur due to different covariates which can be dichotomous or categorical variables such as gender and ethnicity or continuous variables such as age, income, or attitudinal or psychological scales such as motivation. By adding multiple covariates, it becomes much harder for researchers to track the proximal cause of DIF which can be attributed to demographic variables, attitudinal variables, or interactions among them (Tay et al., 2016). Specifically, for a continuous variable, normalization of the variable is usually required which turns a continuous variable into a

categorical variable—age into age group, for example (Kim, Cohen, & Park, 1995). But normalization of a continuous variable is likely to cause loss of information and power for identifying DIF. Use of additional covariates has been employed in recent literature to predict latent information. For example, Tay et al. (2011) proposed an IRT with covariates (IRT-C) model which mimics the process of a multiple-indicators multiple-causes model (MIMIC; Muthen et al., 1991; Wood et al., 2008) that assesses DIF without normalization under a factor analytic framework. The Rasch mixture model (Rost & Davier, 1990; Frick et al., 2014) provides a more general perspective of detecting DIF and splits responses into latent subgroups even when grouping covariates are continuous or unknown. This study aims to assess the robustness of the Rasch mixture model for detecting DIF with respect to the impact of test length, number of latent classes, magnitude of DIF and proportion of DIF items. Overall model fit including Bayesian information criteria (BIC) and Akaike information criteria (AIC) across each scenario are presented and used as evidence for calculating power and false alarm rate or type I error. The empirical distributions of latent class classifier parameter and item difficulty describes their parameter recovery.

Literature Review

This section presents a summary of the literature of the Rasch model with conditional maximum likelihood estimation which is used in the simulation study of this dissertation. The Rasch mixture model function, use of Rasch mixture models, and use of Rasch mixture model in DIF detection are reviewed.

Rasch Model with Conditional Maximum Likelihood Estimation

The mixed Rasch model was introduced by Rost (1990, 1995), and combines a latent trait approach (Rasch, 1960) and latent class approach (general mixture model) to quantitatively model underlying tendency and ability differences. The mixed Rasch model is called the Rasch mixture model in recent studies (e.g., Frick et al., 2014), which not only highlights its relation to the general mixture model but also avoids confusion with mixed effect models. Unlike the LR test, which is a global test for DIF item, Rasch mixture model is capable of detecting DIF at item level by assigning individual item difficulty parameters to different latent classes. Under a Rasch model framework, independence among items is assumed, the probability of observing a vector $y_i = (y_{i1}, \ldots, y_{im})^T$ of the ith subject's responses to all m items can be written as

$$P(Y_i = y_i|\theta_i, b) = \prod_{j=1}^{m} \frac{e^{y_{ij}(\theta_i - b_i)}}{1+e^{(\theta_i - b_i)}} \quad (7)$$

The observed total score, which is the sum of total correct items, is well known as a sufficient statistic for person parameter θ (Rost, 1990). The total score is denoted as $R_i = \sum_{j=1}^{m} Y_{ij}$. Hence The probability function in equation 7 can be summarized as a product of two components

$$P(Y_i = y_i|\theta_i, b) = h(y_i|r_i, b)g(r_i|\theta_i, b) \quad (8)$$

in which the first component h is independent of the person θ and thus θ is conditioned out from likelihood function. The score distribution g should be employed beforehand if the full Rasch likelihood is of interest. Rost and von Davier (1995) suggest employing some distributions for the raw score r_i with a set of auxiliary parameters σ, thus the probability density function can be written as

$$f(y_i|b,\sigma) = h(y_i|r_i,b)g(r_i|\sigma) \quad (9)$$

where σ is the parameter for a raw score distribution. Conditional maximum likelihood (CML) can be used for estimating the item difficulty parameter b by maximuming only the conditional part of h (Frick et al., 2014). Rost (1990) introduces a saturated specification for g which requires $(m-2)$ individual parameters for each possible score, because two extreme scores (r = m and r = 0) were excluded since they contribute nothing to the likelihood. Mean-variance specification, as an alternative to saturated specification, avoided redundancy of too many parameters, and it assigns for g with only two parameters for mean and variance (Rost and van Davier (1995). Frick et al (2014) extended their work and compared three different score specifications: saturated specification, mean-variance specification, and a newly proposed restricted specification; their primary consideration was to reduce the need for too many parameters required by a saturated specification. In this study, a saturated specification for score distribution is employed since simulated sample size is relatively large for substantial amounts of parameters.

Rasch Mixture Model with Expectation-Maximization Algorithm

Generally, a mixture model is a mixture of k components of $f_k(x)$ which can be distributions or models but must come from the same family with same form of parameters that collectively make a mixture model or distribution $f(x)$:

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x) \quad (10)$$

where $0 < \pi_k < 1$ and $\sum_{k=1}^{K} \pi_k = 1$ because the proportions in different classes should add to 1. Given person i belonging to class k, the likelihood function of a Rasch mixture model with K components are described as

$$L(\pi_1, \ldots, \pi_k, b_1, \ldots, b_k, \sigma_1, \ldots, \sigma_k) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \, P(Y_{ij=y_{ij}} | k, b, \sigma) \qquad (11)$$

An expectation-maximization (EM) algorithm is used to estimate parameters in mixture models. Without variables in the data indicating the latent classes to which subjects belong to; it is unlikely to apply maximum likelihood estimation (MLE) directly. EM iteratively approximates weights π_k for each latent class and then re-estimates parameters for Rasch models until convergence is reached. After preliminary estimations for the model parameters for all latent classes, in the E-step probability of expected pattern within each latent class is calculated. In the M-step, model parameters within each latent class are computed by maximum likelihood estimation (MLE). The general estimation process is:

1. Set some initial parameter estimates on the Rasch model.
2. E-step: calculate posterior probabilities π_k for each subject/observation or, in other word, ratio of subject/observation *i*'s contribution to latent class k between *i*'s total contribution.
3. M-step: the calculated posterior distribution in E-step is used as weights of latent classes to re-calculate model parameters using MLE.
4. Repeat E-step and M-step until it reaches convergence.

Summary of Use of the Rasch Mixture Model

A key advantage of the mixture model for detecting DIF is that it can identify test takers for whom the item functions differently once DIF has been identified (Cohen & Bolt, 2005). The Rasch mixture model can serve as an exploratory method for detecting latent classes even when there are no clear covariates such as gender. Compared to a cognitive ability test, responses to a personality or noncognitive test are more likely to be distorted due to factors such as social desirability or malingering, which are usually called faking response style or response distortion. The Rasch mixture model can serve as a good approach to capture diversity of faking response styles (Eid & Zickar, 2007).

The Rasch mixture model has also been applied in assessing dimensionality of scales. For example, Schultz-Larsen et al. (2007) assessed Mini-Mental State Examination (MMSE) which was widely used for detection of cognitive impairment and quantification severity of an individual through a mixed Rasch model item analysis. They concluded a two-dimensional structure of MMSE was a more stable measurement than its original form. Identical parameter estimate across different subgroups can be taken as a criterion for the most important assumption of Rasch model: unidimensionality (Rost et al., 1997). In Rost et al.'s (1997) analysis of two scales of extraversion and conscientiousness (NEO-FFI; Costa & McRae, 1992), they found different solutions: a four-class solution for the extraversion scale and a two-class solution for the conscientiousness scale.

For many educational and psychological research studies, identifying latent class sets the foundation for questioning uses of different cognitive strategies or the tendency

to endorse different test items among different latent classes. Both qualitative approaches (i.e., interview) and quantitative approaches such as analysis of variance (ANOVA) or chi-square tests can used as follow-ups to explore the interaction between latent class and manifest variables, including gender and ethnicity. Izsak et al. (2010) assessed understanding of rational numbers of middle grades teachers. Two different latent classes were detected via use of a Rasch mixture model and interviews were conducted with interested subjects for disclosing latent classes' difference on pedagogical knowledge of rational numbers. The Rasch mixture model is also useful for examining time series data. Cho et al. (2011) combined latent transition analysis (LTA; Collins & Wugalter, 1992) and the Rasch mixture model to analyze cognitive changes over time in the math skill of low-achieving adolescents. They researched the effect of enhanced anchored instruction (EAI; Bottge et al., 2007) over time by revealing how the patterns of latent class classifier changed across different time points. A Rasch mixture model can help reduce the subjectivity built into the common practice of standard setting on performance tests. Jiao et al. (2011) did a simulation study based on the Rasch mixture model to obtain model-based cut scores on the subjects' latent ability scale instead of using a traditional subjective judgmental process of setting cut scores.

There are many applications of the Rasch mixture model across various settings other than education, for example, in sport, exercise, and the motor domains. In Busch and Strauss's (2005) study of Roth's coordination model of movement precision and ability of coordination under pressure, two latent classes were detected for mastering time pressure tasks and they were referred as two distinct strategies: a speed strategy and a

speed-accuracy strategy. The Rasch mixture model has been used in job analysis. Wyse (2018) applied a Rasch mixture model to datasets from multiple professions including bone densitometry, quality management, and cardiovascular interventional radiography and claimed results were informative for developing credentialing exams.

The Rasch mixture model can also be viewed as a multivariate Rasch model. A multivariate Rasch model has been used for scaling and estimation in large-scale educational assessments such as the Programme for International Student Assessment (PISA) and the Trends in Mathematics and Science Study (TIMSS). Compared to other educational measurement settings in which reliable measurement at the test-taker level is the major concern, a fundamental concern in large-scale assessments is to examine interested latent variables' characteristics at population level and relationships among the latent variable and other variables (Adams el al., 2007). The mixed-coefficients multinomial logit (MCML, Adams et al., 1997), another example of a multivariate Rasch model, has a form which is similar to the Rasch mixture model but with more complexity by adding many relationship parameters between θ and item difficulty into the model.

There are various existing fit indices for determining the number of latent classes in a certain Rasch mixture model. Some frequently used measures of fit are Akaike information criterion (AIC), Bayes Information criterion (BIC), and the corrected Akaike information criterion (CAIC). According to Li et al.' (2009) simulation study of the Rasch mixture model, BIC performs better than the rest of the indices for selecting the accurate number of latent classes. In normal practice, various fit indices are provided and compared for the same model.

Use of the Rasch Mixture Model in DIF Detection

There is a growing literature on using the Rasch mixture model as a DIF detection method. Frederick et al. (2010) proposed a Rasch model-based random item mixture model (RIM) with both item difficulty parameters and random person abilities drawn from a univariate normal distribution. They compared DIF detection results to traditional methods (LRT, MH, and ST-p-DIF) and concluded that RIM performs better. Ayrydoust (2015) explored the potential of the Rasch mixture model for detecting DIF with an English as a foreign language (EFL) listening test dataset; the researcher used a neural network method, specifically a classifier, to confirm latent class detection results from fitting a Rasch mixture model and concluded that results were consistent for both approaches. For latent class detection, the Rasch mixture model performs better than a two-parameter or three-parameter IRT mixture model which may result in latent class over-extraction (Alexeev el al., 2011). Time pressure on a performance test could be a cause of DIF for items that are located at the end of the test. Bolt et al. (2002) applied an ordinal constraint by putting items at the end of the original scale at earlier locations. Using a Rasch mixture model, they distinguished two latent classes of test takers: a "speeded" class and a "non-speeded" class. In particular, they used a Markov Chain Monte Carlo (MCMC) estimation algorithm for fitting the Rasch mixture model instead of an EM algorithm.

In a traditional practice of comparing subgroups, manifest variables—often demographic factors such as gender, ethnicity and socio-economic status--are used to categorize groups within populations. However, this may be inappropriate for explaining

cognitive processes of test takers because manifest variables are chosen according to researchers' conjectures (Cohen & Bolt, 2005). Subgroups which are characterized by manifest variables may still be heterogeneous since the variables which cause the within-group heterogeneity remain unnoticed (Maij-deMeij et al., 2008). A Rasch mixture model instead can reveal DIF, but the researcher should be cautious about interpreting the characteristics of the identified latent classes. Effects of heterogeneity within latent classes has been discussed in Wu and Huang's (2010) study assessing the Beck Depression Inventory-II-Chinese version (BDI-II-C). In their study, 10 items displayed uniform DIF when they used a 2-class partial credit model (PCM) to fit the data. They claimed that construct validity held because there was no DIF detected within two latent classes after DIF items were assigned different parameters value for each latent group.

Model identification constraints are used when comparing potential latent classes. By using a simulation approach to examine parameter recovery and correct model detection rate, Wu and Paek (2018) compared a conventional constraint which set the ability mean of latent classes to be equal and an anchor item constraint which used class-invariant items, and concluded that there was high agreement between these two constraints across multiple simulated conditions.

Simulation has been used increasingly in recent literature for assessing robustness of the Rasch mixture model for DIF detection. Table 1 summarizes simulation studies of DIF detection using the Rasch mixture model. The accuracy of fitting a Rasch mixture model is determined by various aspects, which includes proportion of replications that recover the true latent structure, parameter recovery of latent class classifier and item

difficulty, and proportion of replications that catch true DIF. The most common purpose in these literatures was to examine the influence of a covariate which is also commonly referred as collateral information or effects of a manifest variable on detecting DIF items using a Rasch mixture model (Dai, 2013; DeAyala et al., 2002; Li et al., 2016; Samuelsen, 2005). One of the biggest challenges of using a Rasch mixture model for detecting DIF is that it is difficult to interpret the qualitative meaning of the identified latent groups. One solution is to include an auxiliary variable or covariate such as gender, age, or income. For example, Smit, Kelderman, and Flier (1999) found a decrease in standard errors and easier assignment of latent subgroups when a covariate was included.

There are two ways of connecting a covariate in Rasch mixture modeling: (1) First, manipulation of a binary covariate (which could be gender, for example, in real practice) to overlap with the proportion of latent classes. For instance, a 50/50 covariate split overlapped with a 30/70 latent class split, say 50% of sample was male and 50% was female, in which 30% of male and 70% of female belonged to a latent class and 70% of male and 30% of female went into another latent class. (2) Second, use a covariate to predict parameters of the Rasch mixture model. Dai (2013) used logistic regression to link a binary covariate and probability of latent group classifier in which the covariate was the predictor and the logit of latent group classifier probability was the outcome, and thus turned the Rasch mixture model into a hierarchical structure. The author claimed that inclusion of a collateral variable helps increase the correct rate of latent structure identification. Li et al. (2016) extended Dai's (2013) work by adding an additional continuous covariate to predict latent ability.

Markov chain Monte Carlo estimation dominates this area of simulation since the majority of existing simulation studies on DIF detection using a Rasch mixture model fell into a Bayesian framework. WinBUGS is the most commonly used software, but usage of R is growing in more recent literature. The eight simulation studies listed in Table 1 assessed robustness of the Rasch mixture model for detecting DIF when there were two latent classes. Latent ability of latent class came from either an identical standard normal distribution or two normal (also known as the Gaussian) distributions with different means but the same standard deviation of 1.0. The proportion of latent class classifier was another common manipulated factor: an equal size (50% - 50%) or an unequal size of latent classes. A 30% - 70% divide of latent class was the most frequent unequal size setting. The proportion of DIF items and the DIF size were the most crucial factors in this kind of simulation. For item difficulty of the reference latent class, researchers either simulated data from a normal distribution or a uniform distribution or used a fixed symmetrical array (Frick et al., 2014). Different magnitudes of DIF (i.e., 0.3) were then added to certain proportion of reference latent class item difficulties and the rest remained the same for both latent classes. In this way, proportion of DIF items and magnitude of DIF were manipulated to build several different DIF patterns. In addition to the above common manipulated factors, different model settings (Frick et al., 2014; Li et al., 2016) and different missing data types (Li et al., 2016) have also been examined but still need further research. Simulation conditions varied from study to study. All the eight simulation studies listed in Table 1 used dichotomous items. The sample size for each replication varied from 500 to 3,000. Each sample per replication was divided into a focal

group or reference group if there was a binary covariate (i.e., gender) included. Number of replications per scenario ranged from 11 to 500.

Table 1
Simulation Studies of Factors affecting Rasch Mixture Model Outcomes

DeAyala et al. (2002)	1PL mixture model with one binary covariate (authors claimed using a 2PL but constrained discriminant parameter to be 1)	Monte Carlo study using MULTILOG (Thissen, 1991), BILOG (Mislevy & Bock, 1990), EQUATE (Baker, 1993) and IRTDIF (KIM & Cohen, 1992)	1. Theta: from normal distribution *N* (-1, 1) and *N* (0, 1) 2. Proportion of latent class: 17% - 83%, 30% - 70% and 50% - 50% 3. DIF items: 0%, 10% and 30% with two levels of DIF: 0.3 and 1.0 4. Item difficulty for reference latent class: from normal distribution *N* (0, 1)	1. Test length: 30 dichotomous items 2. Sample size: 3,000 (500 for focal group and 2,500 for reference group) 3. 50 replications each condition 4. Number of LC: 2	Rasch mixture model accuracy increased as proportion in latent class got closer to 0.5.
Samuelsen (2005)	Assessing effectiveness of Rasch mixture model on DIF detection when latent classes overlapped with one binary covariate (manifest group)	Markov chain Monte Carlo using WinBUGS	1. Theta: from normal distribution *N* (0, 1) and *N* (-1, 1) for each latent class 2. Proportion of latent class: equal size (50% - 50%) and unequal (20% - 80%), each latent class overlap with a manifest variable in five conditions: 100%, 90%, 80%, 70% and 60% 3. DIF items: 2 (10%), 6 (30%) and 10 (50%) items with three levels of DIF: 0.4, 0.8 and 1.2 4. Item difficulty of reference latent class: from	1. Test length: 20 dichotomous items 2. Sample size: 500 and 2,000 3. 100 replications per condition 4. Number of LC: 2	Accuracy of Rasch mixture model increased as the overlap between latent class and the manifest variable increased.

			a uniform distribution *unif* (-2, 2)		
Frederickx et al. (2010)	Random Item Mixture model	Markov chain Monte Carlo using WinBUGS	1. Theta: identical distribution from a normal distribution N (0, 1) and different normal distributions from *N* (0, 1) and *N* (.5, 1) 2. Proportion of latent class: equal size 3. DIF item: 0 and 5(with item difficulty difference: .4, .6, .8, -.8 and -1) 4. Item difficulty of reference class: from a *unif* (-1, 1)	1. Test length: 20 and 50 dichotomous items 2. Sample size: 500 and 1,000 3. 20 replications each condition 4. Number of LC: 2	RIM is better than traditional DIF detection methods: LRT, MH, and STD *p*-DIF.
Preinerstorfer & Formann (2012)	Parameter recovery and model selection of the Rasch mixture model	Conditional maximum likelihood estimation with *mRm* package in R	1. Theta: from a normal distribution *N* (0, 1) 2. Proportion of latent class: equal size and unequal (25% - 75%) 3. Binary responses were simulated from a Bernoulli distribution 4. Item difficulty for reference latent class: from a *unif* (-2, 2)	1. Test length: 10, 15, 25 and 40 dichotomous items 2. Sample size: 500, 1,000 and 2,500 3. 200 replications each scenario 4. Number of LC: 1 and 2	Parameter estimate accuracy of the Rasch mixture model increased as sample size and number of items increased; parameters were more precisely estimated for medium range

					parameters and in homogeneous group (1 latent class)
Dai (2013)	Rasch mixture model with one binary covariate which predicted latent class classifier	Markov chain Monte Carlo using WinBUGS and SAS was used to generate datasets	1. Theta: identical distribution from a normal distribution *N* (0, 1) and different normal distributions from *N* (0, 1) and N (1, 1); 2. Proportion of latent class: 15% - 85%, 30% - 70% and 50% - 50%; 3. DIF items: 20% (6) and 40% (12) 4. Covariate effect (3 levels)	1. Test length: 30 dichotomous items 2. Sample size: 1,000 3. 11 replications for each condition 4. Number of LC: 2	Collateral information (covariate) has positive effect on detecting latent structure.
Frick et al. (2014)	Assessing three Rasch mixture model settings on detecting DIF: the saturated model, the mean-variance model, and the restricted model	Conditional maximum likelihood estimation via EM algorithm using *psychomix* package in R	1. Theta: two latent groups from different normal distributions *N* (-x/2, 0.3) and *N* (x/2, 0.3), where x from [0, 4] in steps of 0.4 2. Proportion of latent class: equal size (50% - 50%) 3. DIF item: 2 items with DIF of -y and y, where y from [0, 4] in steps of 0.2 4. Item difficulty of reference latent class: from	1. Test length: 20 dichotomous items 2. Sample size: 500 3. 500 replications each condition 4. Number of LC: 2	Pros and cons of different Rasch mixture model settings were discussed across various scenarios

			[-1.9, 1.9] with increments of 0.2		
Choi et al. (2016)	Assessing performance of four information criteria (AIC, AICC, BIC and SABIC) on DIF detection using Rasch mixture model	Maximum likelihood parameter estimation (MLR) using Mplus	1. Theta: fixed (1 level): one class from a normal distribution *N* (0, 1) and one from *N* (.5, 1) 2. Proportion of latent class: equal size (50% - 50%) and unequal size (70% - 30%) 3. DIF item: 30% (9), 60% (18) and 90% (27) with 4 levels of DIF: 0.5, 0.75, 1.0 and 1.5 4. Three patterns of DIF: fully crossing, gradually decreasing and fully parallel; reference latent group item difficulty from a normal distribution N (0, 1)	1.Test length: 30 dichotomous items 2. Sample size: 3,000 3. 100 replications for each condition 4. Number of LC: 2	Four information criteria should combine for selecting best model.

Li et al. (2016)	Rasch mixture model with one binary covariate which predicted latent class classifier, and a continuous covariate which predicted latent ability	Markov chain Monte Carlo using *R2WinBUGS* package in R	1. Theta: from a normal distribution *N* (0, 1) for one latent class and a normal distribution *N* (1, 1) for the other 2. Proportion of latent class: 50% - 50% and 30% - 70% with binary covariate (30% - 70%) 3. Correlation between the continuous covariate and latent trait: two levels (0.2 and 0.8) 4. DIF items: two levels of average DIF (1.5 and 1) 5. Item difficulty for reference latent class: from a normal distribution *N* (0, 1) 6. 3 types of missing data 7. Correlation between dichotomous covariate and classifier: 2 levels (odds ratio = 10 and 1)	1. Test length: 30 dichotomous items 2. Sample size: 2,000 3. 25 replications for each condition 4. Number of latent classes: 2	Accuracy of latent group classification, model parameter recovery, and overall model fit were discussed across several simulated conditions

Problem and Purpose

Measures of constructs, including both cognitive and non-cognitive constructs, are associated with making decisions about a person. It is of great importance to ensure the differences identified through items reflecting a construct only attribute true difference to subgroups, which is referred as measurement invariance. Measurement invariance influences reliability and validity of an effective test and it should be confronted at both the test level and the item level. Differential item functioning occurs when there is difference in the probability of endorsing an item across different subgroups and the difference is conditioned on a continuous latent trait. The presence of DIF to a large extent threatens fairness and thus incurs bias in a measurement process. DIF items should be eliminated or fixed before implementation of a test to the targeted population. Various parametric and non-parametric (e.g., MH, LRT) methods have been proposed with statistical tests and applications for detecting DIF. Those methods commonly use manifest variables or covariates to categorize elements of a population or of a collected sample into subgroups for DIF identification. However, it is problematic to assume homogeneity within subgroups categorized by observed covariates. Latent class analysis, as an alternative approach to observed covariate-based subgroups, may be a preferable way to discern homogenous subgroups from a measurement perspective.

The Rasch mixture model (RMM), which is also known as mixed effect model or mixture Rasch model in varied literatures, has the capacity of extracting latent classes and detecting DIF item in a test. RMM allocates different item parameters for each identified latent class and the difference among each latent class' item parameters serve as evidence

for flagging DIF items. There are numerous studies of the Rasch mixture model from applications in various fields to simulation across multiple scenarios. Simulation is a powerful tool for assessing the robustness of the Rasch mixture model since it can mimic varied possible situations simultaneously by manipulating several factors. To date, most of simulation studies of using the Rasch mixture model for detecting DIF focus on a test length from 20 items to 30 items or even 50 items (Friderickx et al., 2010). And only one (Preinerstorfer & Formann, 2012) of eight simulation studies listed in Table 1 included a condition with test length of 10 items, though the primary focus of Preinerstorfer and Formann's research was on parameter recovery instead of DIF. Through both live-testing results and simulation, Weiss (1982) indicated that there was no reduction in the quality of measurement after reducing test length. Long test length is more likely to incur fatigue for the test taker, missing data, and random responses to items at the end of the test. Additionally, no simulation study was located on DIF detection using a Rasch mixture model examining the condition of three latent classes. Most previous simulation studies in this area used Markov chain Monte Carlo from a Bayesian framework perspective.

Given inadequate information from previous simulation studies on uniform DIF detection using the Rasch mixture model, this dissertation addressed: (1) how a three LC structure interacts with other manipulated factors including test length, proportion of DIF items, and magnitude of DIF, with outcomes of correct model selection and item parameter recovery; (2) accuracy of the Rasch mixture model in detecting DIF for a short length test; (3) correct model or latent structure classification rate and model parameter recovery of the Rasch mixture model using an expectation-maximization algorithm with

conditional maximum likelihood estimation; and (4) comparison between a three LC structure and a two LC structure in recovery of true DIF effect size using the RMM.

Glossary of Terms

Item Response Theory (IRT)

Compared to classical test theory which focuses on test level, item response theory models test takers' performance at the individual item level. The core idea of IRT is that probability of a correct or desired response of an item is a mathematical function of person and item parameters. Those parameters include latent ability/attitude of a test taker, item difficulty, item discrimination, and pseudo-guessing.

1 PL IRT model. IRT model with a person parameter and one item parameter: latent ability and item difficulty. Specifically, the 1 PL model is sample independent in which the rank of all items is the same for all test takers despite of person ability and the rank of person ability is independent of item difficulty. The feature only applies to 1 PL IRT model.

2 PL IRT model. IRT model with a person parameter and two item parameters: latent ability, item difficulty and item discrimination.

3 PL IRT model. IRT model with a person parameter and three item parameters: latent ability, item difficulty, item discrimination and pseudo-guessing.

Rasch Model (RM)

Rasch model has same mathematical form as the 1PL IRT model including the sample independence or sample-free feature. However, the RM should be viewed as a different theory conceptualizing the relation between data and modeling. RM emphasizes

the superiority of the model and misfitting items and misfitting persons will be excluded from fitting a Rasch model.

Rasch Mixture Model (RMM)

The Rasch mixture model is a combination of the Rasch model and a mixture model. In a mixture model, a distribution f is a mixture of K component distribution of $f_1, f_2, \ldots f_k$ if $f(x) = \sum_{k=1}^{K} \lambda_k f_k(x)$ with λ_k being the mixing weights, $\lambda_k > 0, \sum_k = 1$. A Rasch mixture model is commonly used for identifying latent classes within a targeted population.

Latent Class (LC)

A latent class is an unobserved homogenous subgroup which has item parameters for some items distinct from those for another latent class.

Monte Carlo method

The Monte Carlo method uses randomness through repeated sampling to solve complicated problems which are often difficult or impossible to solve via other approaches. Monte Carlo is a practice to draw a large number of samples from a certain distribution, then calculate parameters of interest from those samples.

Item Response Function (IRF)

Probability function of correct or desired response to a test item under an IRT framework.

Differential Item Functioning (DIF)

Differential item functioning occurs when there is difference in the probability of a response to an item for different subgroups, excluding the effect of true differences in

latent ability. DIF should be distinguished from true latent ability difference which is referred to as impact in the IRT literature. Without a special statement, DIF in this study refers to uniform DIF.

Uniform DIF. Uniform DIF between two subgroups on an item is invariant across the latent trait continuum or in other words independent of latent ability.

Non-uniform DIF. Non-uniform DIF is different probability of response between groups at different location on a latent trait continuum.

Expectation-Maximization (EM) Algorithm

Expectation-maximization algorithm is a method to find maximum likelihood estimates of parameter of a statistical model which depends unobserved latent variables (Dempster et al., 1977).

Parameter Recovery

Parameter recovery is used to describe how the mean of estimated parameters after multiple replications comes closer to the true value.

Akaike Information Criterion (AIC)

Akaike information criterion is a model selection for comparing the relative quality of models for a given set of data. The lowest AIC is preferred.

Bayesian Information Criterion (BIC)

Bayesian information criterion is also a model selection method which takes number of data points into account compared to AIC. The lowest BIC is preferred when applying to comparing models for a given dataset.

Chapter Two: Method

Introduction

This chapter focuses on mapping the flow of the study which includes listing fixed factors and manipulated varying factors, explaining the data generation process, and clarifying model performance procedures involving the criteria used in analyzing simulation results across various simulation conditions. The goal of setting conditions in the simulation was to represent real world measurement scenarios.

Simulation Design

The core idea of this simulation is the combination of variables contributing to fitting a Rasch mixture model. Through multiple replications for each condition in this Monte Carlo study, various empirical distributions can be generated as evidence of Rasch mixture model parameter recovery and power for detecting DIF. Factors are categorized into two types: fixed factors and varying factors (also known as manipulated factors in most simulation studies). The selection of fixed factors and varying factors is considered according to the research interest of this dissertation and to build on past research with DIF in a Rasch mixture model. Fixed factors consist of number of replications for each condition, the difference between latent ability of latent groups (impact), and sample size for each replication. Varying factors are test length, number of latent classes, proportion of cases in latent classes, proportion of DIF items, and DIF pattern. Overall, varying factors produce 2 (test length) * 2 (number of latent classes) * 2 (proportion of cases in

latent classes) * 3 (proportion of DIF items) * 2 (DIF patterns) = 48 different conditions. Table 2 shows the summary of all simulated conditions across all manipulated factors. There were 100 replications for each scenario on latent structure recovery and 200 replications for each scenario on parameter recovery.

Table 2
Summary of Conditions across Five Factors

			Test Type					
N of LC	**P of LC**	**DIF Type**	10 items, 2 DIF items	10 items, 4 DIF items	10 items, 6 DIF items	30 items, 6 DIF items	30 items, 12 DIF items	30 items, 18 DIF items
Two LC	N: 1,500, 1,500	S						
		G						
	N: 2,000, 1,000	S						
		G						
Three LC	N: 1,000, 1,000, 1,000	S						
		G						
	N: 1,500, 1,000, 500	S						
		G						

Note. S = symmetric DIF pattern; G = gradient DIF pattern; LC = latent class; N = sample size.

Fixed Factors

Number of Replications. A Markov chain Monte Carlo (MCMC) method has been commonly used in most of the simulation studies using a Rasch mixture model to detect DIF and WinBUGS is the commonly used software for MCMC. WinBUGS is a flexible but time-consuming statistical software (Samuelsen, 2005). As a result, the number of replications of MCMC studies on this topic was relatively small, usually around 30 or even 11 (Dai, 2013). In contrast, simulations studies using conditional maximum likelihood estimation with an expectation-maximization (EM) algorithm make use of a larger number of replications: Preinerstorfer and Formann (2012) used 200

replications for each condition and Frick et al. (2014) employed 500 replications. This Monte Carlo simulation employed 100 replications for each manipulated scenario on latent structure recovery and 200 replications for each manipulated scenario on parameter recovery and uses an EM algorithm for parameter estimation. Datasets for Rasch mixture model fitting replications were generated through the *mirt* package in R. Details for generating a dataset for each individual replication are in the following "data generation process" section.

Impact. Impact refers to the true mean difference on the latent trait between two latent classes. It has been concluded by several previous studies that larger group differences resulted in more accurate model estimation (DeMars & Lau, 2011; Lubke & Muthén, 2007; Lu & Jiao, 2009). Since the influence of impact is not the focus of this dissertation, a medium level of impact is assumed for better model accuracy. In a two LC structure, one group has latent mean of 0 and the other has 1.0; in a three classes latent structure, three groups have latent mean of -1.0, 0 and 1.0, respectively.

Sample Size. Several studies found that a larger sample size resulted in increased Rasch mixture model accuracy, including latent structure recovery and item parameter recovery (Dai, 2013; Frick et al., 2014; Preinerstorfer & Formann, 2012; Samuelsen, 2015). Additionally, a larger sample size increased the speed of RMM convergence (Frick et al., 2014) and reduced the confounding influence of sampling leading to unstable model parameter estimation (Choi et al., 2016). Samuelsen (2005) and Li et al. (2016) included a sample size of 2,000 in their research, and DeAyla et al. (2012) and Choi et al. (2016) employed sample sizes of 3,000 for each replication within each

individual simulated cell. All of them got relatively high model convergence rates and stable parameter recovery. Since the impact of sample size variation is not of interest in this dissertation, a sample size of 3,000 for each replication per condition was used to ensure stable and accurate model recovery.

Varying Factors

Test Length. What is the proper test length for a scale? The question has received extensive study in the past decades. There is no agreement on this issue because it is influenced by various factors such as content area, type of scale, and data collection processes. From a qualitative perspective, it largely depends on the content area and the targeted test taker group. In practice, short scales are effective from the perspective of collecting data without respondent fatigue--some are less than 10 items (PHQ-9, Kroenke, Spitzer & Williams, 2001) or even just one item such as the net promoter score (NPS, Mortimer, 2008) for assessing customer experience. A psychological personality identification scale or a performance test is longer, the Beck Depression Inventory (Byrne et al, 1994) with 21 items as an example, as the mindset of psychology test takers is quite different from that of a customer. Educational assessments, especially for those measuring students' performance, usually have a test length longer than 30 items. For example, the Fall 2008 grade 5 Michigan science assessment (Li, Hong, & Lissitz, 2014) has 45 items; the ACT mathematics test has 60 items (act.org). However, a longer cognitive test is more likely to cause fatigue for the test taker and may yield random answers to items at the end of the test. From a quantitative perspective, a longer test length benefits speed of Rasch mixture model convergence and accuracy of model

estimation (Preinerstorfer & Formann, 2012). This simulation manipulated test length: 10 items and 30 items, which are close to real research settings for a brief test and a longer attitude or personality measure yet shorter than a standardized educational assessment.

Proportion of Cases in Latent Classes. There are two types of proportions of cases in latent classes: an equal size design and an unequal size design. In the current study, for the equal size design, groups in a two LC structure have groups size of 1,500 and 1,500 and groups in three LC structure have sample sizes of 1,000, 1,000 and 1,000. For the unequal size design, groups in a two LC structure have sample size of 1,000 and 2,000 and groups in three latent structure have sample size of 500, 1,000 and 1,500.

Number of Latent Classes. Two levels were used: a two LC structure (k = 2) and a three classes structure (k = 3). The latent class with latent mean $\overline{\theta}$ = 0 was set to be the reference latent class LC_r. The remaining two latent classes were set to be focus latent classes with notations: LC_{f_1} and LC_{f_2} with $\overline{\theta}$ = 1 and $\overline{\theta}$ = -1, respectively. The reference latent class always has the largest sample size in an unequal sample design.

Proportion of DIF items. There were three levels of proportions of DIF items: 20%, 40% and 60%, which account for 2, 4 and 6 items for a 10-item test and 6, 12, 18 for a 30-item test.

DIF ($\boldsymbol{\Delta_b}$) Pattern. Item difficulty parameters b_{LCr} for the reference latent class LC_r which has latent mean $\overline{\theta}$ = 0 are simulated from a uniform distribution $unif(-1, 1)$. An array of DIF are then added to DIF items in the first focal latent classes (LC_{f_1}) and the remaining item difficulties remain the same with a corresponding part of the reference latent class: $b_{LC_{f1}} = b_{LC_r} + \Delta_b$. The item parameter for the second focal latent class

(LC_{f_2}) is created by adding a double magnitude of Δ_b: $b_{LC_{f2}} = b_{LC_r} + 2\Delta_b$. The detailed information of Δ_b is listed in Table 3. There are two types of DIF patterns in this simulation study: symmetric and gradient. A symmetric DIF pattern ensures that the overall item difficulty remains the same for each latent group or, in other words, there is no differential test functioning (DTF) because each item DIF is set to be cancelled out in this design (see Figure 1). The biggest advantage of a symmetric DIF simulates a real measurement situation when absence of DTF disguises the presence of item level DIF. The second is a gradient DIF pattern. Instead of adding a symmetric array of Δ_b it adds an array of gradually changed DIF effect sizes which have same positive direction to focus latent classes (see Figure 2).

Both symmetric and gradient DIF pattern designs can be used to examine recovery of a set of different magnitudes through the Rasch mixture model simultaneously.

Table 3

List of Δ_b for Different Types of Tests

	DIF Type	
Test Type	Symmetric	Gradient
10 items, 2 DIF items	(-1.8, 1.8, 0, 0, 0, 0, 0, 0, 0 ,0)	(2.0, 1.0, 0, 0, 0, 0, 0, 0, 0 ,0)
10 items, 4 DIF items	(-1.8, -0.9, -0.9, 1.8, 0, 0 ,0, 0, 0, 0)	(2.0, 1.5, 1.0, 0.5, 0, 0, 0, 0, 0 ,0)
10 items, 6 DIF items	(-1.8, -1.2, -0.6, 0.6, 1.2 ,1.8, 0, 0 ,0 ,0)	(2.0, 1.7, 1.4, 1.1, 0.8, 0.5, 0, 0, 0, 0)
30 items, 6 DIF items	(-1.8, -1.2, -0.6, 0.6, 1.2 ,1.8, 0, 0, … 0, 0)	(2.0, 1.7, 1.4, 1.1, 0.8, 0.5, 0, … 0, 0, 0)
30 items, 12 DIF items	(-1.8, -1.5, -1.2, -0.9, -0.6, -0.3, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 0, 0, ... 0, 0)	(2.0, 1.8, 1.6, 1.4, 1.2, 1.0, 0.8, 0.6, 0.4, 0.3, 0.2, 0.1, 0, 0, ... 0, 0)
30 items, 18 DIF items	(-1.8, -1.6, -1.4, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 0, 0, ... 0, 0)	(1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0, … 0, 0)

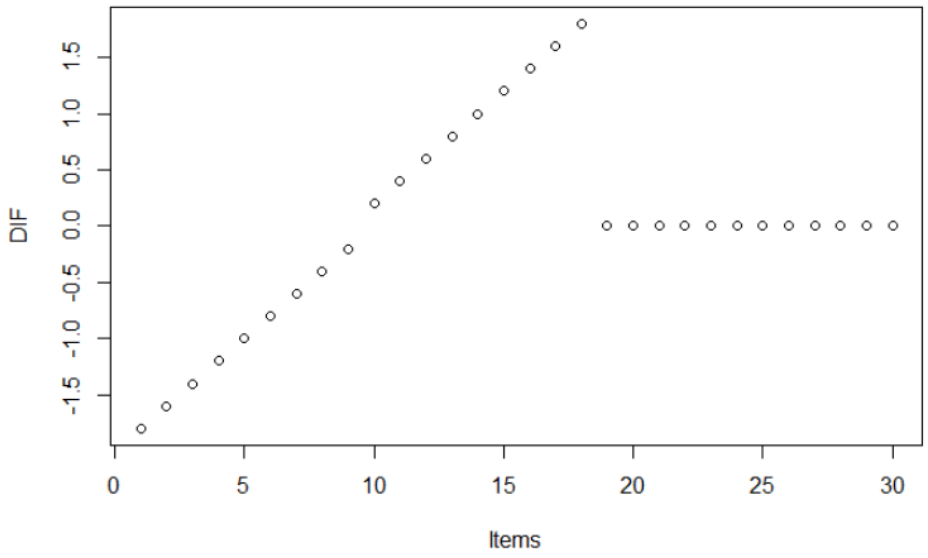

Figure 1
A Symmetrical DIF Pattern

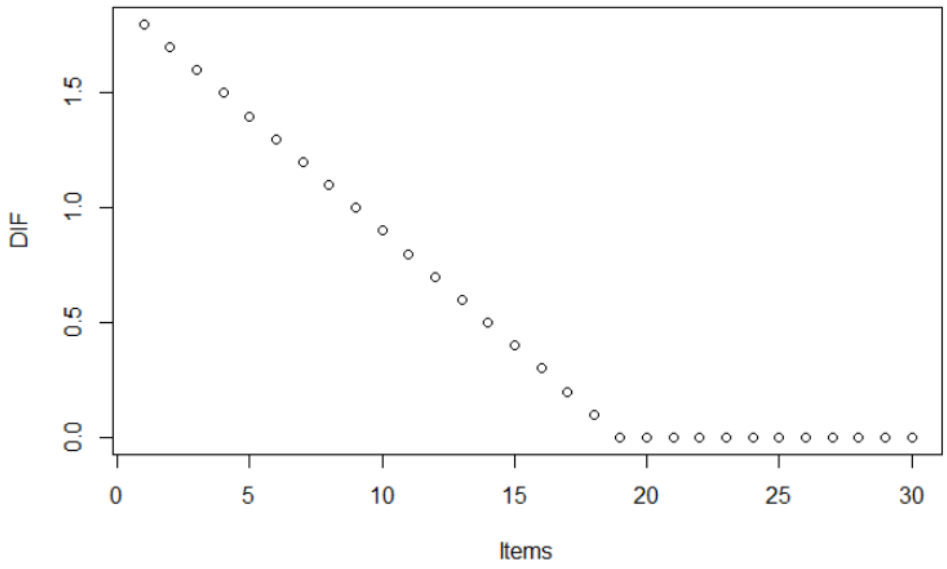

Figure 2
A Gradient DIF Pattern

Data Generation Process

R, a programming language with strength in simulating data and flexibility, is used as a primary tool for data generation and manipulation. Latent ability θ_{LC_r} is drawn from a standard normal distribution $N(0, 1)$ for the reference latent class LC_r based on a sample size for a certain condition, and then corresponding sample size of latent ability $\theta_{LC_{f1}}$ for focal latent class LC_{f_1} is drawn from a normal distribution with mean of 1 and standard deviation of 1: $N(1, 1)$. In the three LC structure, corresponding sample size of latent ability $\theta_{LC_{f2}}$ for the second focal latent class is drawn from $N(-1, 1)$. Item difficulty parameters (b_{LC_r}) for the reference latent class are drawn from a uniform distribution *unif* (-1, 1), then corresponding Δ_b are added to b_{LC_r} to create $b_{LC_{f1}}$ and $b_{LC_{f2}}$. Binary responses based on generated θ and b are simulated for each latent class through the *mirt* package in R. After this step, the number of datasets equals the number of simulated latent classes. In each dataset, a column (i.e., variable) named true_lc is used for marking true (i.e., simulated) classifier for each response. A dataset for a replication within a certain cell is created through merging datasets for every latent classes. The order of responses (i.e., rows in a dataframe) are shuffled. Then the merged and shuffled dataset is ready for model fitting using a Rasch mixture model.

Performance Analysis

The purpose of performance analysis is to determine how well generated parameters are recovered from analyses of simulated datasets and investigate interactions among manipulated varying factors. Model recovery consists of two parts: latent structure recovery and parameter recovery. Efficiency of model convergence is an aspect of

importance for a simulation study and it is reflected in running time for a model fitting cycle. Running time for 200 replications of each manipulated scenario on parameter recovery was recorded, which can be serve as an indicator of efficiency of fitting a Rasch mixture model via the EM algorithm. Main effects for five manipulated factors (number of items, proportion of DIF items, LC structure, group size and DIF patterns) and interactions among them on latent structure recovery and DIF recovery were examined by four analysis of variance tests.

Latent Class Structure Recovery

Rasch mixture models were used for fitting simulated datasets across manipulated situations for parameters of an assumed number of latent classes $k = 1, 2, 3, 4$. The best fitting model was selected based on which model had minimum values of information criteria. The model information criteria used in this simulation are commonly used model selection indices: Akaike information criterion (AIC, Equation 12) and Bayesian information criterion (BIC, Equation 13). Both AIC and BIC select the more correct model by introducing a penalty term for number of parameters in a model. The selected Rasch mixture model has a smaller an information criterion value. There are several literatures suggesting that BIC is better statistic on DIF detection using the Rasch mixture model than AIC (e.g., Li et al., 2009). Latent structure recovery rate is calculated from the number of correct selected models divided by number of replications for each condition. For example, when fitting a Rasch mixture to a two LC structure using BIC as model information criterion, the latent structure recovery rate is calculated as the number

of replications in which k = 2 associated with minimum BIC divided by the total number of replications in each cell, which is 100 in this study.

$$AIC = -2ln(\hat{L}) + 2P \tag{12}$$

where $\hat{L}$ is the maximized value of the likelihood of the model and P is the number of estimated parameters in the model.

$$BIC = -2ln(\hat{L}) + Pln(N) \tag{13}$$

N is the number of observations.

Parameter Recovery

Parameter recovery contains two parts: recovery of classifier for different latent classes and recovery of item difficulty. Classifier parameter indicates proportion of each latent class. Mean squared error (MSE, Equation 14) and root mean square error (RMSE, Equation 15) are used for assessing overall recovery of different magnitudes of DIF among latent classes across 48 different manipulated situations. RMSE is also used to examine overall performance classifier parameter recovery using RMM. RMSE is squared root result of MSE. A smaller MSE or a smaller RMSE indicates a better parameter recovery for a true DIF among latent classes. MSE is used for explaining the transformation of RMSE and only RMSEs are used for ANOVAs on LC structure recovery and parameter recovery.

$$MSE = \sum_{i=1}^{n} \frac{(\widehat{\Delta_b} - |\Delta_b|)^2}{n} \tag{14}$$

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\widehat{\Delta_b} - |\Delta_b|)^2}{n}} \tag{15}$$

where $\widehat{\Delta_b}$ is predicted magnitude of DIF for each item of a test, Δ_b is corresponding true DIF, and n is number of test items for the cell. RMSE of overall DIF for each replication and RMSE of overall classifier parameter for each replication are used as dependent variable of ANOVAs assessing the main effects and interactions effects for five manipulated factors.

Label Switching Problem

Label switching is a quite common and problematic issue when estimating parameters from mixture models. In this study label switching means that the empirical distribution of parameters from replications is invariant when switching component labels (i.e., the label for latent classes). For example, we simulate an item with DIF of -1.8 in a dataset with two latent classes. The output order of the two latent classes (i.e., components in EM output) is random, thus results in an unknown direction of the DIF item. The solution in this study is to take the absolute value for each item difficulty difference across latent classes in each replication. The label switching problem happens in the classifier parameter π_k too. The solution is to extract π_k based on relative magnitudes among latent classes.

Number of Replications and Running Time for Each Simulation Scenario

The number of replications for latent structure recovery scenarios and number of replications for item DIF recovery scenarios are fixed. Since four types of the number of latent classes (k = 1 to k = 4) are analyzed, the simulation time for latent structure recovery is four times than that for parameter recovery. Thus, the number of replications for each latent structure recovery situation was set to be lower than that for the parameter

recovery situation: 100 replications for each cell in Table 2 for latent structure recovery simulation and 200 replications for each cell in Table 2 for parameter recovery simulation. The running time for 200 replications within each individual simulation condition was recorded as an evidence of effectiveness of using a Rasch mixture model for DIF detection.

Analysis of Variance (ANOVA)

There were four ANOVAs conducted in this study for examining main and interaction effects of five manipulated variables: number of items (i.e., test length), proportion of DIF, DIF pattern, group size, and LC structure.

Using RMSE of DIF logit value from each replication as an outcome variable and factors as independent variables, an analysis of variance was conducted to explore the effects of five factors and interactions among them on overall DIF recovery. To assess the main effects and interaction effects of five manipulated factors on classifier parameter recovery, RMSE of the classifier parameter from each replication was used as the dependent variable for the ANOVA on classifier parameter recovery.

In order to examine the main effects and interaction effects of five manipulated factor on latent structure recovery, another two ANOVA tests were conducted. AIC and BIC were normalized by taking natural log of each. One of these two ANOVA tests uses *ln* (AIC) as a dependent variable and the other uses *ln* (BIC) as a dependent variable. The reason for using different ANOVAs for AIC and BIC was the formulation of AIC and BIC differs and so both could not be included as DVs in a single analysis.

Instead of using statistical significance test for the factor effects, partial eta-squared (η^2, see Equation 15) was used as a measure of effect size of main effects and interactions for the five manipulated factors. As the sample size was large, statistical significance was deemed a less useful assessment than effect size. As suggested by Cohen (1988), a rule of thumb was employed to assess η^2 effect size: small when $\eta^2 \leq 0.06$, medium when $0.06 < \eta^2 \leq 0.14$ and large when $\eta^2 > 0.14$.

$$\eta^2 = SS_{effect}/SS_{total} \tag{15}$$

where SS_{effect} is the sum of squares for a factor and SS_{total} is sum of squares for all effects. Only main effect and interaction with effects size $\eta^2 > 0.01$ were regarded as interpretable and so small effect sizes were considered interpretable. Comparison of means was conducted for different levels of involved factors when interactions had an effect size $\eta^2 > 0.01$.

Software and Packages

R (R Core Team, 2019) is the primary tool for simulation and visualization in this dissertation. The R package *mirt* was used for data generation and the R package *psychomix* for fitting Rasch mixture models. Specific codes for whole simulation procedure including data generation, visualization, latent class structure recovery and parameter recovery can be found in Appendix A.

Chapter Three: Results

Introduction

This chapter describes the simulation results for all 48 designed scenarios. At the test level, latent structure recovery using the Rasch mixture model is summarized. At the item level, parameter recovery, which includes classifier parameter and DIF, are examined. Multiple tables and figures are presented for clarification. Findings are synthesized at the conclusion of this chapter.

The following naming rule for each simulation condition was used: a four-digit name was used to define test length and the number of items with true DIF not equal to 0. Lowercase letters s and g followed by a four-digit number indicate whether DIF items are in a symmetric DIF pattern or a gradient DIF pattern. Lc2 and lc3 followed by an underscore sign describe the true latent structure. Lowercase letters e and u are used to describe whether a simulation condition has an equal group or an unequal latent group design. For instance, 1002s_lc2_e refers to a 10-item measure with 2 DIF items in a symmetric pattern with 2 LC of equal group size, and 3006g_lc3_u refers to a 30-item with 6 DIF items in a gradient pattern test which has three LC of unequal group size.

Since four Rasch mixture models (k = 1 to k = 4) were used for fitting simulated datasets, the workload for latent structure recovery took nearly four times longer than that for parameter recovery. From this consideration, 100 replications were used for latent structure recovery for each scenario which was lower than the 200 replications for

parameter recovery. Although a lower number of replications were used for latent structure recovery, it cost about 75 hours to complete compared to 26 hours for parameter recovery simulations.

Latent Structure Recovery

Latent structure recovery refers to the rate of accuracy of the Rasch mixture model for picking the true LC structure which was two LC and three LC in this study. Each of the 48 simulated datasets was fitted by four Rasch mixture models with the assumed number of LC K from 1 to 4 and other model settings controlled to be the same. The model selection information criterion was used for deciding which of the four models performed the best for the simulated dataset. The latent structure recovery rate was calculated by the number of selected K divided by the number of replications which was 100 for each simulation scenario.

The Akaike information criterion (AIC, Equation 12) and Bayesian information criterion (BIC, equation 13) were used in this study where the lowest value of the information criterion indicated the best model fit. Model fit rate results are presented across different test types and using different information criteria from Table 4 to Table 7. Columns headings for the correct structure are marked in bold and italic font in each table while the highest recovery rate per condition is bolded within the table.

Latent structure recovery results for the 10-item test with a two LC structure are shown in Table 4. AIC based recovery rates were higher than 0.85 except for 1002g_lc2_e and 1002g_lc2_u which have AIC based latent structure recovery rates of 0.62 and 0.37. However, BIC-based model recovery rate was quite low except for

1006s_lc2_e and 1006s_lc2_u with rates of 0.92 and 0.79. BIC tended to favor k = 1 while the true structure was k = 2. For both AIC and BIC, latent structure recovery rate for the symmetric DIF pattern was higher than that for the gradient DIF pattern. AIC performed similarly between situations with equal group size and unequal group size. BIC showed a higher rate for situations with equal group sizes than for situations with unequal group sizes.

Table 4
10 Item Two LC Structure Recovery Proportions

			Equal Group Size				Unequal Group Size			
Test Type	**DIF Pattern**		*k = 1*	***k = 2***	*k = 3*	*k = 4*	*k = 1*	***k = 2***	*k = 3*	*k = 4*
1002	Symmetric	AIC	0.00	***0.85***	0.15	0.00	0.05	***0.81***	0.13	0.01
		BIC	***0.72***	0.28	0.00	0.00	***0.90***	0.10	0.00	0.00
	Gradient	AIC	0.16	***0.62***	0.20	0.02	***0.47***	0.37	0.16	0.00
		BIC	***1.00***	0.00	0.00	0.00	***1.00***	0.00	0.00	0.00
1004	Symmetric	AIC	0.00	***0.91***	0.08	0.01	0.00	***0.95***	0.05	0.00
		BIC	0.31	***0.69***	0.00	0.00	***0.60***	0.40	0.00	0.00
	Gradient	AIC	0.02	***0.85***	0.13	0.00	0.04	***0.87***	0.08	0.01
		BIC	***0.96***	0.04	0.00	0.00	***1.00***	0.00	0.00	0.00
1006	Symmetric	AIC	0.00	***0.89***	0.08	0.03	0.00	***0.92***	0.07	0.01
		BIC	0.08	***0.92***	0.00	0.00	0.21	***0.79***	0.00	0.00
	Gradient	AIC	0.00	***0.85***	0.12	0.03	0.01	***0.86***	0.13	0.00
		BIC	0.49	***0.50***	0.00	0.01	***0.86***	0.14	0.00	0.00

Note. k = 2 was the true number of LC.

Latent structure recovery results for the 30-item test with a two LC structure is shown in Table 5. Overall performance using AIC and BIC was lower compared to that for the 10-item test with the two LC structure. For AIC, the model recovery rate was higher for situations with unequal group sizes than that with an equal group size. For BIC, the model recovery rate was still quite low except for 3012s_lc2_e (0.71), 3018s_lc2_e (0.75) and 3018s_lc2_u (0.75). Similar to that for the 10-item test with two LC structure, BIC tended to favor k = 1. But AIC seemed inconclusive between the true

LC structure k = 2 and k = 4. BIC was higher than AIC in scenario 3012s_lc2_e (BIC: 0.71, AIC: 0.37), 3018s_lc2_e (BIC: 0.75, AIC: 0.41), and 3018s_lc2_u (BIC: 0.75, AIC: 0.52). AIC-based model recovery rates were higher for the situation with unequal group size while BIC-based model recovery rates were similar for equal and unequal group sizes. Both AIC- and BIC-based LC structure recovery rates were higher for the scenario with a symmetric DIF pattern than for that with a gradient DIF pattern.

Table 5
30 Item Two LC Structure Recovery Proportions

			Equal Group Size				Unequal Group Size			
Test Type	**DIF Pattern**		*k = 1*	***k = 2***	*k = 3*	*k = 4*	*k = 1*	***k = 2***	*k = 3*	*k = 4*
3006	Symmetric	AIC	0.00	0.38	0.05	***0.57***	0.00	***0.56***	0.05	0.39
		BIC	***0.76***	0.01	0.02	0.21	***0.85***	0.00	0.00	0.15
	Gradient	AIC	0.00	0.25	0.07	***0.68***	0.02	***0.51***	0.03	0.44
		BIC	***0.65***	0.00	0.04	0.31	***0.83***	0.00	0.00	0.17
3012	Symmetric	AIC	0.00	0.37	0.00	***0.63***	0.00	0.44	0.05	***0.51***
		BIC	0.02	***0.71***	0.02	0.25	0.25	***0.48***	0.02	0.25
	Gradient	AIC	0.00	0.22	0.10	***0.68***	0.01	0.40	0.10	***0.49***
		BIC	***0.54***	0.01	0.08	0.37	***0.73***	0.00	0.05	0.22
3018	Symmetric	AIC	0.00	0.41	0.01	***0.58***	0.00	***0.52***	0.08	0.40
		BIC	0.00	***0.75***	0.02	0.23	0.00	***0.75***	0.05	0.20
	Gradient	AIC	0.00	0.20	0.06	***0.74***	0.00	0.48	0.03	***0.49***
		BIC	0.19	0.18	0.08	***0.55***	***0.67***	0.04	0.03	0.26

Note. *k* = 2 was the true number of LC.

Latent structure recovery results for the 10-item test with a three LC structure is shown in Table 6. There is no model recovery rate more than 0.4 for either AIC or BIC and BIC is lower than AIC. 1006s_lc3_e (0.4) and 1006s_lc3_u (0.39) showed the highest model recovery rate. In this table, AIC and BIC showed some degree of agreement on choosing k = 2 when the true LC structure was k = 3. But BIC once again was more conservative than AIC, with BIC having some substantial probabilities of picking k = 1. AIC was higher than BIC but cannot be used as evidence for finding the

correct LC structure since recovery rates were so low. Neither AIC nor BIC are suggested as a model selection method for detecting DIF using a Rasch mixture model for the 10-item test with three LC.

Table 6
10 Item Three LC Structure Recovery Proportions

			Equal Group Size				Unequal Group Size			
Test Type	**DIF Pattern**		*k = 1*	*k = 2*	***k = 3***	*k = 4*	*k = 1*	*k = 2*	***k = 3***	*k = 4*
1002	Symmetric	AIC	0.00	***0.91***	0.07	0.02	0.01	***0.79***	0.16	0.04
		BIC	0.19	***0.81***	0.00	0.00	0.39	***0.61***	0.00	0.00
	Gradient	AIC	0.00	***0.90***	0.08	0.02	0.01	***0.81***	0.16	0.02
		BIC	0.30	***0.70***	0.00	0.00	***0.80***	0.20	0.00	0.00
1004	Symmetric	AIC	0.00	***0.69***	0.27	0.04	0.00	***0.78***	0.20	0.02
		BIC	0.01	***0.99***	0.00	0.00	0.05	***0.95***	0.00	0.00
	Gradient	AIC	0.00	***0.88***	0.09	0.03	0.00	***0.83***	0.15	0.02
		BIC	0.04	***0.96***	0.00	0.00	0.21	***0.79***	0.00	0.00
1006	Symmetric	AIC	0.00	***0.52***	0.40	0.08	0.00	***0.47***	0.39	0.14
		BIC	0.00	***1.00***	0.00	0.00	0.00	***1.00***	0.00	0.00
	Gradient	AIC	0.00	***0.74***	0.23	0.03	0.00	***0.85***	0.14	0.01
		BIC	0.01	***0.98***	0.01	0.00	0.04	***0.96***	0.00	0.00

Note. *k* =3 was the true number of LC.

Latent structure recovery results for the 30-item test with a three LC structure is shown in Table 7. Compared to the scenario with the 10-item tests with a two LC structure, the AIC-based model recovery rate increased; but BIC still was low with nearly all model recovery rates equal to 0.00 for the 3-class structure. There were four AIC-based LC structure recovery rates exceeding 0.50 and they were 3012s_lc3_e (0.73), 3012s_lc3_e (0.53), 3018s_lc3_e (0.85) and 3018s_lc3_u (0.81). Both AIC and BIC demonstrated a similar conservative model selection pattern as in the simulation conditions of 10 items tests with a two LC structure.

Table 7
30 Item Three LC Structure Recovery Proportions

			Equal Group Size				Unequal Group Size			
Test Type	**DIF Pattern**		*k = 1*	*k = 2*	***k = 3***	*k = 4*	*k = 1*	*k = 2*	***k = 3***	*k = 4*
3006	Symmetric	AIC	0.00	***0.77***	0.17	0.06	0.00	***0.84***	0.13	0.03
		BIC	0.00	***0.98***	0.00	0.02	0.20	***0.79***	0.00	0.01
	Gradient	AIC	0.00	***0.84***	0.11	0.05	0.00	***0.74***	0.08	0.18
		BIC	0.00	***1.00***	0.00	0.00	0.08	***0.85***	0.00	0.07
3012	Symmetric	AIC	0.00	0.11	***0.73***	0.16	0.00	0.34	***0.53***	0.13
		BIC	0.00	***0.96***	0.01	0.03	0.00	***0.99***	0.00	0.01
	Gradient	AIC	0.00	***0.63***	0.11	0.26	0.00	***0.61***	0.12	0.27
		BIC	0.00	***0.98***	0.00	0.02	0.00	***0.95***	0.00	0.05
3018	Symmetric	AIC	0.00	0.01	***0.85***	0.14	0.00	0.05	***0.81***	0.14
		BIC	0.00	***0.95***	0.00	0.05	0.00	***0.97***	0.00	0.03
	Gradient	AIC	0.00	0.36	0.24	***0.40***	0.00	***0.41***	0.19	0.40
		BIC	0.00	***0.89***	0.00	0.11	0.00	***0.87***	0.02	0.11

Note. $k = 3$ was the true number of LC.

Analysis of Variance (ANOVA) on Latent Structure Recovery

Two analysis of variance tests were run for examining the main effects and interactions of five manipulated factors on latent structure recovery rate. *ln* (AIC) and *ln* (BIC) were used as the dependent variables in the ANOVAs. The natural log form of AIC and BIC was used for normalizing AIC and BIC since the relative magnitude of an information criterion was of interest for model selection. These five factors were number of test items (2 levels), proportion of DIF items (3 levels), DIF pattern (2 levels), group size (2 levels), and LC structure (2 levels).

By comparing the results of the two ANOVAs, both main and interaction effects were quite similar in effect size except for the main effect size of LC structure, with $\eta^2 = 0.03$ for *ln* (AIC) as the dependent variable and $\eta^2 = 0.01$ for *ln* (BIC) as the dependent variable.

ANOVA with ln (AIC) as Dependent Variable: The independence and normality assumptions of analysis of variance were met. The homogeneity of variance assumption was violated ($p < 0.01$) for number of items, proportion of DIF, DIF type, LC structure, and group size, but analysis of variance is robust with respect to violation of homogeneity of variance with a balanced design.

All five main effects were found to have interpretable effect sizes with $\eta^2 > 0.01$ (Table 8). For number of test items, F (1,4752) = 1,016,082.52, η^2 = .995, with a lower mean *ln* (AIC) for 10-item tests (mean = 10.43, SD < 0.01) than that for 30-item tests (mean = 11.49, SD < 0.01). This result was expected since AIC value depends on the number of items, with lower values for shorter tests. For DIF type, F (1,4752) = 249.64, η^2 = 0.05, with a lower mean *ln* (AIC) for gradient pattern tests (mean = 10.95, SD < 0.01) than that for symmetric pattern tests (mean = 10.97, SD < 0.01). For LC structure, F (1,4752) = 125.28, η^2 = 0.03, with a lower mean *ln* (AIC) for three LC structure (mean = 10.96, SD < 0.01) than that for two LC structure (mean = 10.97, SD < 0.01). For group size, F (1,4752) = 593.80, η^2 = 0.03, with a lower mean *ln* (AIC) for equal group size (mean = 10.95, SD < 0.01) than that for unequal group size (10.97). For proportion of DIF items, F (2,4752) = 106.79, η^2 = 0.04, Tukey's HSD was used to assess the group differences for proportion of DIF and all three pairs of group differences were found to be statistically significant. Mean *ln* (AIC) for 60% DIF was 10.95 with SD < 0.01, for 40% DIF 10.96 with SD < 0.01, and for 20% DIF 10.97 with SD < 0.01.

Table 8

Summary Table for Effects of Five Manipulated Factors on ln (AIC)

Source	Sum of Squares	df	Mean Square	F	*p*	η^2
n_of_items	1332.667	1	1332.667	1016082.520	<.001	.995
p_of_DIF	.280	2	.140	106.788	<.001	.043
DIF_type	.327	1	.327	249.642	<.001	.050
LC_structure	.164	1	.164	125.280	<.001	.026
group_size	.779	1	.779	593.795	<.001	.111
n_of_items * p_of_DIF	.002	2	.001	.823	.439	< .001
n_of_items * DIF_type	.025	1	.025	19.079	<.001	.004
n_of_items * LC_structure	.001	1	.001	.529	.467	< .001
n_of_items * group_size	.001	1	.001	1.020	.313	< .001
p_of_DIF * DIF_type	.062	2	.031	23.698	<.001	.010
p_of_DIF * LC_structure	.021	2	.010	7.957	<.001	.003
p_of_DIF * group_size	.010	2	.005	3.956	.019	.002
DIF_type * LC_structure	.099	1	.099	75.597	<.001	.016
DIF_type * group_size	.001	1	.001	.877	.349	< .001
LC_structure * group_size	.001	1	.001	.928	.335	< .001
n_of_items * p_of_DIF * DIF_type	.013	2	.006	4.942	.007	.002
n_of_items * p_of_DIF * LC_structure	.002	2	.001	.637	.529	< .001
n_of_items * p_of_DIF * group_size	8.241E-006	2	4.120E-006	.003	.997	< .001
n_of_items * DIF_type * LC_structure	.009	1	.009	6.693	.010	.001
n_of_items * DIF_type * group_size	< .001	1	< .001	< .001	.995	< .001
n_of_items * LC_structure * group_size	< .001	1	< .001	.300	.584	< .001
p_of_DIF * DIF_type * LC_structure	.008	2	.004	3.186	.041	.001
p_of_DIF * DIF_type * group_size	.007	2	.003	2.604	.074	.001
p_of_DIF * LC_structure * group_size	.002	2	.001	.609	.544	< .001

DIF_type * LC_structure * group_size	.027	1	.027	20.207	< .001	.004
n_of_items * p_of_DIF * DIF_type * LC_structure	.004	2	.002	1.644	.193	.001
n_of_items * p_of_DIF * DIF_type * group_size	.005	2	.003	1.928	.146	.001
n_of_items * p_of_DIF * LC_structure * group_size	.003	2	.001	1.030	.357	< .001
n_of_items * DIF_type * LC_structure * group_size	.002	1	.002	1.563	.211	< .001
p_of_DIF * DIF_type * LC_structure * group_size	.003	2	.002	1.262	.283	.001
n_of_items * p_of_DIF * DIF_type * LC_structure * group_size	.006	2	.003	2.264	.104	.001
Error	6.233	4752	.001			
Total	1340.765	4799				

Only the interaction between DIF type and LC structure was found to be interpretable with $\eta^2 = 0.02$. Simple effects analyses were used to examine difference between two levels of DIF type at each level of LC structure. Figure 3 shows that the *ln* (AIC) difference between two LC structure and three LC structure is larger at symmetric DIF level than at gradient DIF level. The means and SDs of DIF pattern by LC structure are shown in Table 9.

Symmetric tests with two LC structure had the lowest mean *ln* (AIC) = 10.98 (SD < 0.01). Gradient tests with two LC structure had mean *ln* (AIC) = 10.95 (SD < 0.01). Symmetric tests with three LC structure had mean *ln* (AIC) = 10.96 (SD < 0.01). Gradient tests with three LC structure had mean *ln* (AIC) = 10.95 (SD < 0.01).

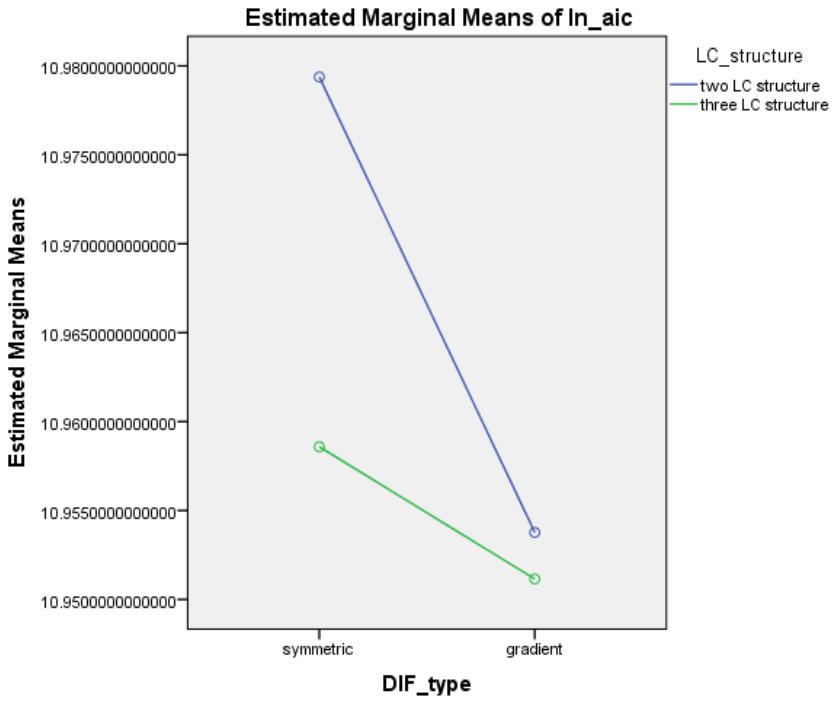

Figure 3
Plot for Mean Difference of ln (AIC) for DIF Type by LC Structure Interaction

Table 9
Means and SDs of ln (AIC) for DIF Type by LC Structure Interaction

DIF Type	LC Structure	Mean	SD
Symmetric	Two LC	10.98	.001
	Three LC	10.96	.001
Gradient	Two LC	10.95	.001
	Three LC	10.95	.001

ANOVA with ln (BIC) as Dependent Variable: Independence and normality assumptions of analysis of variance were met. The homogeneity of variance assumption was violated ($p < 0.01$) for number of items, proportion of DIF, DIF type, LC structure and group size, but analysis of variance is robust with respect to violation of homogeneity of variance with a balanced design.

All five main effects were found to have interpretable effect sizes with $\eta^2 > 0.01$. For number of test items, F (1,4752) = 1034694.768, η^2 = .995, with a lower mean *ln* (BIC) for 10-item tests (10.44) than that for 30-item tests (11.50). For DIF type, F (1,4752) = 250.78, η^2 = 0.05, with a lower mean *ln* (BIC) for gradient pattern tests (10.95) than that for symmetric pattern tests (10.97). For LC structure, F (1,4752) = 63.49, η^2 = 0.01, with a lower mean *ln* (BIC) for the three LC structure (10.96) than that for the two LC structure (10.97). For group size, F (1,4752) = 593.70, η^2 = 0.11, with a lower mean *ln* (BIC) for equal group size (10.96) than that for unequal group size (10.98). For proportion of DIF items, F (2,4752) = 106.74, η^2 = 0.04, Tukey's HSD was used to assess the group differences for proportion of DIF and all three pairs of group differences were found to be statistically significant. Mean *ln* (BIC) for 60% DIF was 10.96, for 40% DIF 10.97, and for 20% DIF 10.98.

Table 10

Summary Table for Effects of Five Manipulated Factors on ln (BIC)

Source	Sum of Squares	df	Mean Square	F	*P*	η^2
n_of_items	1334.959	1	1334.959	1034694.768	< .001	.995
p_of_DIF	.275	2	.138	106.738	< .001	.043
DIF_type	.324	1	.324	250.779	< .001	.050
LC_structure	.082	1	.082	63.491	< .001	.013
group_size	.766	1	.766	593.701	< .001	.111
n_of_items * p_of_DIF	.002	2	.001	.812	.444	< .001
n_of_items * DIF_type	.025	1	.025	19.192	< .001	.004
n_of_items * LC_structure	< .001	1	< .001	.282	.595	< .001
n_of_items * group_size	.001	1	.001	1.036	.309	< .001
p_of_DIF * DIF_type	.061	2	.031	23.791	< .001	.010
p_of_DIF * LC_structure	.020	2	.010	7.867	< .001	.003
p_of_DIF * group_size	.010	2	.005	3.955	.019	.002
DIF_type * LC_structure	.098	1	.098	75.924	< .001	.016
DIF_type * group_size	.001	1	.001	.854	.355	< .001
LC_structure * group_size	.001	1	.001	.841	.359	< .001
n_of_items * p_of_DIF * DIF_type	.013	2	.006	4.963	.007	.002
n_of_items * p_of_DIF * LC_structure	.002	2	.001	.637	.529	< .001
n_of_items * p_of_DIF * group_size	8.685E-006	2	4.342E-006	.003	.997	< .001
n_of_items * DIF_type * LC_structure	.009	1	.009	6.626	.010	.001
n_of_items * DIF_type * group_size	1.731E-007	1	1.731E-007	< .001	.991	< .001
n_of_items * LC_structure * group_size	< .001	1	< .001	.297	.586	< .001
p_of_DIF * DIF_type * LC_structure	.008	2	.004	3.208	.041	.001
p_of_DIF * DIF_type * group_size	.007	2	.003	2.594	.075	.001
p_of_DIF * LC_structure * group_size	.002	2	.001	.613	.542	< .001

DIF_type * LC_structure * group_size	.026	1	.026	20.164	< .001	.004
n_of_items * p_of_DIF * DIF_type * LC_structure	.004	2	.002	1.641	.194	.001
n_of_items * p_of_DIF * DIF_type * group_size	.005	2	.002	1.931	.145	.001
n_of_items * p_of_DIF * LC_structure * group_size	.003	2	.001	1.029	.357	< .001
n_of_items * DIF_type * LC_structure * group_size	.002	1	.002	1.549	.213	< .001
p_of_DIF * DIF_type * LC_structure * group_size	.003	2	.002	1.254	.285	.001
n_of_items * p_of_DIF * DIF_type * LC_structure * group_size	.006	2	.003	2.270	.103	.001
Error	6.131	4752	.001			
Total	1342.846	4799				

Only the interaction between DIF type and LC structure was found to be interpretable with $\eta^2 = 0.02$. Simple effects analyses were used to examine difference between two levels of DIF type at each level of LC structure. Figure 4 shows that mean *ln* (BIC) is larger for the two LC structure than three LC structure at symmetric DIF level while it is smaller for the two LC structure than three LC structure at gradient DIF level. The means and SDs of DIF pattern by LC structure are showed in Table 11.

Symmetric tests with two LC structure got lowest mean *ln* (BIC) = 10.99 (SD < 0.01). Gradient tests with two LC structure had mean *ln* (BIC) = 10.96 (SD < 0.01). Symmetric tests with three LC structure had mean *ln* (BIC) = 10.97 (SD < 0.01). Gradient tests with three LC structure had mean *ln* (BIC) = 10.96 (SD < 0.01).

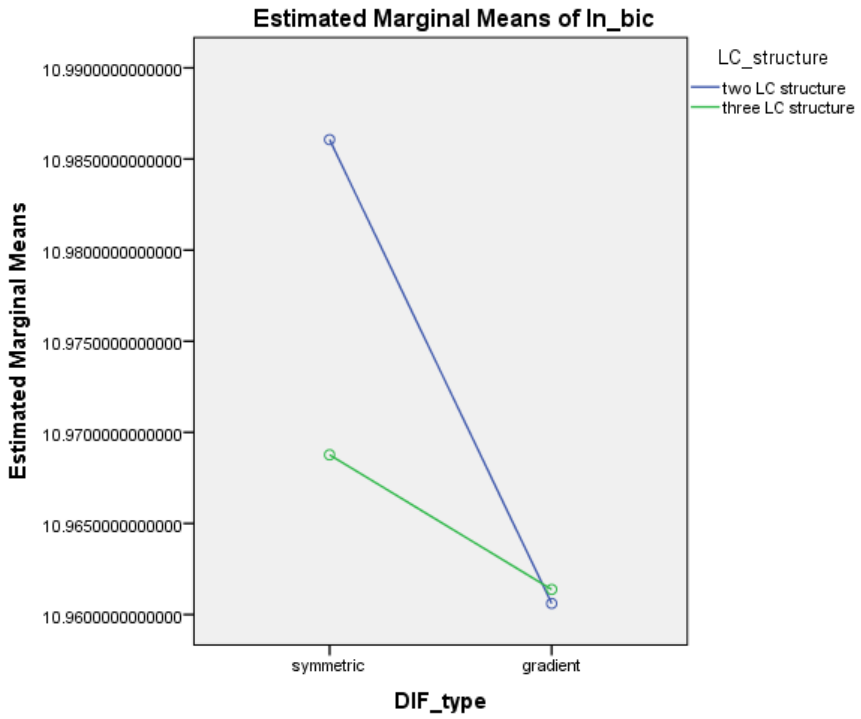

Figure 4
Plot for Mean Difference of ln (BIC) for DIF Type by LC Structure Interaction

Table 11
Means and SDs of ln (BIC) for DIF Type by LC Structure Interaction

DIF Type	LC Structure	Mean	SD
Symmetric	Two LC	10.99	.001
	Three LC	10.97	.001
Gradient	Two LC	10.96	.001
	Three LC	10.96	.001

Parameter Recovery

Parameter recovery consists of two parts: classifier (i.e., LC indicator parameter) recovery π_k and item level DIF recovery Δ_b. Comparisons were conducted based on manipulated factors: test length, proportion of DIF, types of DIF pattern, and type of latent structure which included a different number of LC and a different size of LC. Total parameter recovery simulations for 48 situations took 26 hours. Layouts of tables and figures are organized to reflect comparisons' logic across different manipulated factors.

Classifier Parameter Recovery

There are 12 figures with the same layout for classifier parameter recovery in this section, and an ANOVA is conducted afterwards to assess the main effects and interactions effects of the five manipulated factors on classifier parameter recovery. Each figure describes classifier parameter recovery for a test type which includes four kinds of LC group sizes. The top two plots of each figure are for simulated datasets which have two LC in which the left one has LC of equal size (1,500 and 1,500) while the right plot has LC of unequal size (2,000 and 1,000). The lower two plots of each figure are for simulated datasets which have a three LC structure. The lower left plot is for the simulated dataset with three LC of equal size (1,000, 1,000 and 1,000) and the lower right plot is for the simulated dataset with three LC of unequal size (1,500, 1,000, and 500).

Uppercase P refers to the classifier parameter π_k; the true probability for each LC is indicated in the title of each plot. Mean P refers to the mean probability of the LC with 200 replications. Points of probabilities of the classifier for each scenario are connected into lines in order to show fluctuations of the classifier parameters for the replications.

For the two LC structure, a green horizontal line is added to mark the true proportion of LC and a red horizontal line marks the mean of the classifier parameter across 200 replications. For the three LC structure, the green solid line and green dotted line, respectively, indicate the LC with largest true proportion and smallest true proportion among three LC; the red solid line and red dotted line indicate the mean of the largest proportion and the mean of the smallest proportion among three LC respectively.

Classifier parameter recovery for test 1002s: for the 1002s_lc2_e scenario, the mean probability of the LC with a larger proportion was 0.52 (SD = 0.02), which nearly overlapped with the true proportion. For 1002s_lc2_u, the mean was 0.52 with SD 0.03, but it failed to recover the true probability of 0.67.

For 1002s_lc3_e, the mean for the LC with the largest proportion was 0.40 (SD = 0.04) and the mean for the LC with the smallest proportion was 0.28 (SD = 0.03). For 1002s_lc3_u, the mean for the LC with the largest proportion was 0.39 (SD = 0.04) and the mean for the LC with the smallest proportion was 0.29 (SD = 0.03).

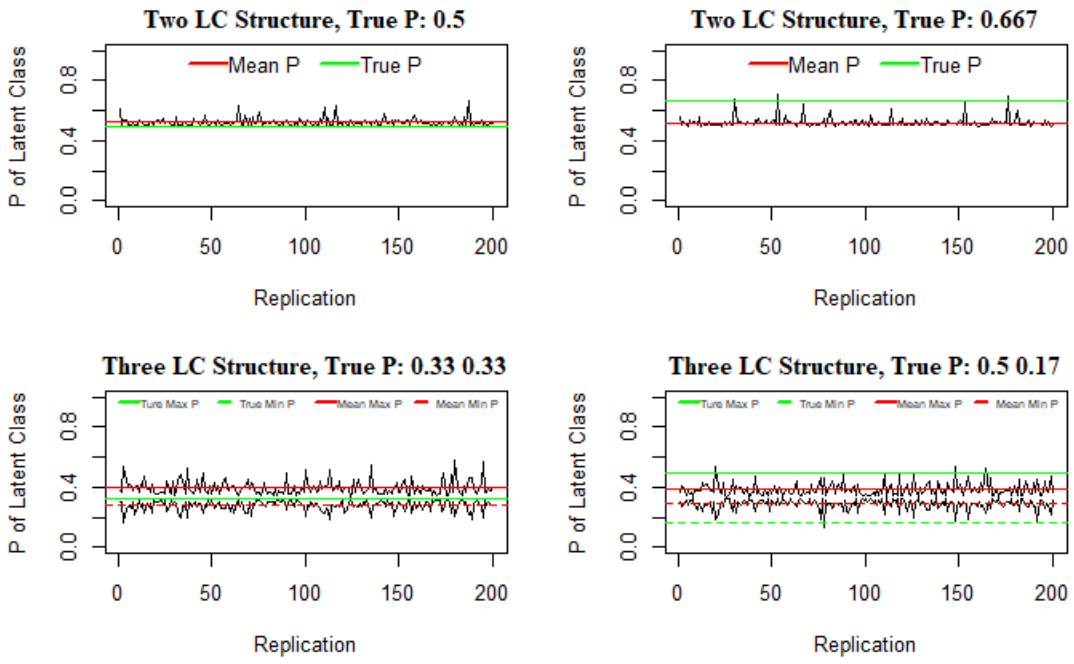

Figure 5
Classifier Parameter Recovery for Test 1002s

Classifier parameter recovery for test 1002g: for the 1002g_lc2_e scenario, the mean probability of the LC with the larger proportion was 0.52 (SD = 0.02), which nearly overlapped with the true proportion. For 1002g_lc2_u, the mean was 0.52 with SD 0.01, which failed to recover the true probability of 0.67.

For 1002g_lc3_e, the mean for the LC with the largest proportion was 0.40 (SD = 0.04) and the mean for the LC with the smallest proportion was 0.28 (SD = 0.03). For 1002g_lc3_u, the mean for the LC with the largest proportion was 0.38 (SD = 0.04) and the mean for the LC with the smallest proportion was 0.29 (SD = 0.03).

There was nearly no difference between the 1002s and 1002g test type. For both test types, there were acceptable classifier recoveries in situations with the equal group

design. However, classifier recoveries in situations with unequal LC sizes were much worse.

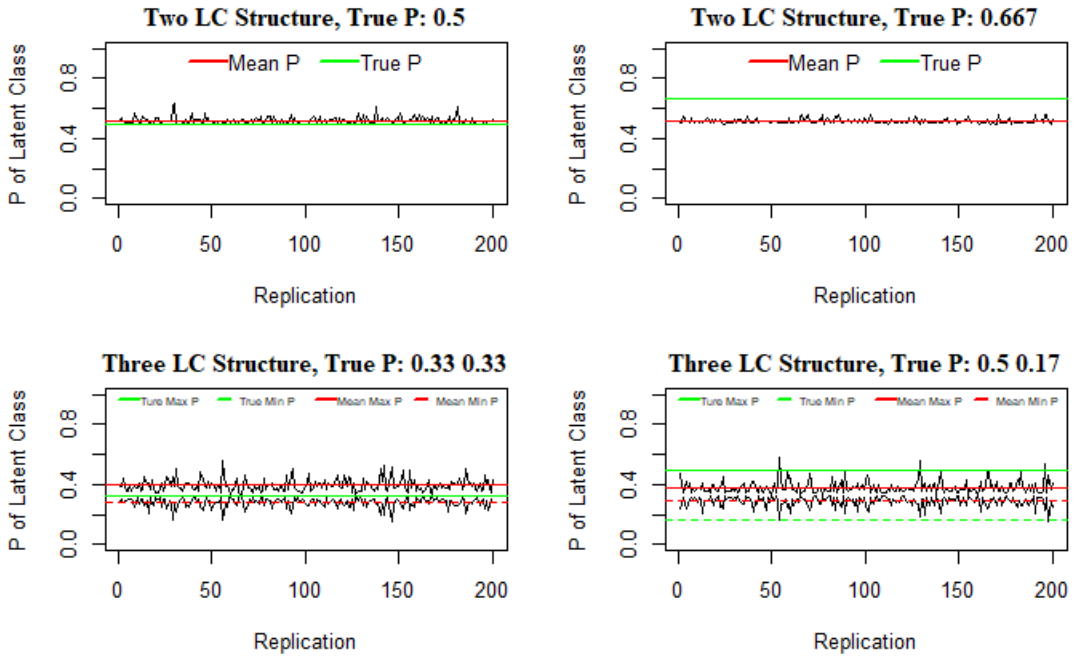

Figure 6
Classifier Parameter Recovery for Test 1002g

Classifier parameter recovery for test 1004s: the mean of the classifier parameter for 1004s_lc2_e was 0.52 (SD = 0.03) and the mean of the classifier parameter for 1004s_lc2_u was 0.55 (SD = 0.06) which was slightly higher than 1004s_lc2_e. For the three LC structure with an equal group design, the mean of the classifier parameter for the LC with the larger proportion was 0.41 (SD = 0.05) and for the LC with the smaller proportion was 0.27 (SD = 0.03). For the three LC structure with unequal size design (1004s_lc3_e), the mean of the classifier parameter for the LC with the larger proportion was 0.39 (SD = 0.05) and for that with the smaller proportion was 0.28 (SD =

0.04). Although the true proportions were quite different between the two LC3 situations, classifier parameter recoveries were quite similar.

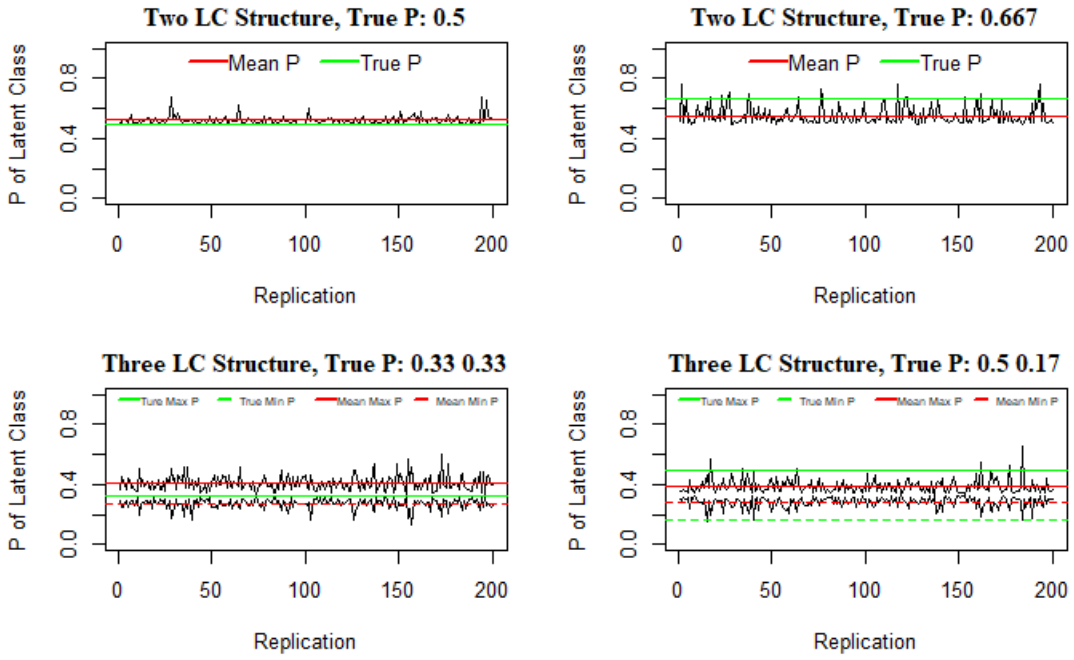

Figure 7
Classifier Parameter Recovery for Test 1004s

Classifier parameter recovery for test 1004g: the mean of the classifier parameter for 1004g_lc2_e was 0.52 (SD = 0.02) and the mean of the classifier parameter for 1004g_lc2_u was 0.52 (SD = 0.02). For the three LC structure with an equal group design (1004g_lc3_e), the mean of the classifier parameter for the LC with the larger proportion was 0.40 (SD = 0.05) and for the LC with the smaller proportion was 0.28 (SD = 0.04). For the three LC structure with an unequal size design (1004g_lc3_u), the mean of the classifier parameter for the LC with the larger proportion was 0.39 (SD = 0.04) and for that with the smaller proportion was 0.28 (SD = 0.04).

Although the true proportions were quite different between the LC2 and LC3 designs, the classifier parameter recoveries were quite similar for test type 1004g.

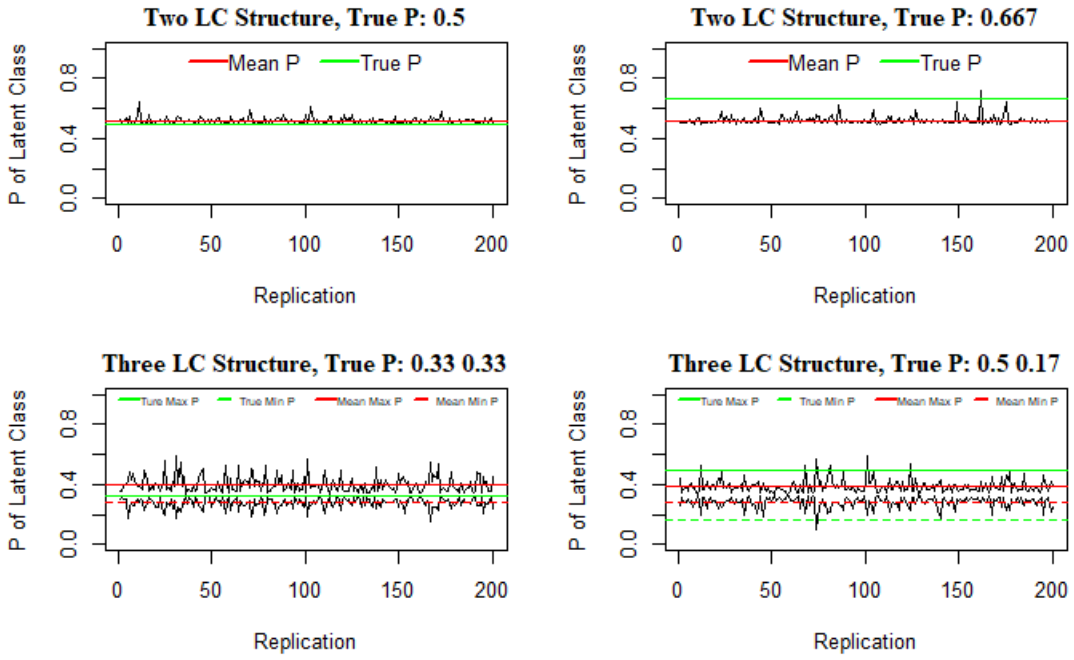

Figure 8
Classifier Parameter Recovery for Test 1004g

Classifier parameter recovery for test 1006s: the mean of the classifier for 1006s_lc2_e was 0.52 with SD of 0.02 and the mean of the classifier for 1006s_lc2_u was 0.58 with SD of 0.06. The classifier parameter recovery showed some higher value for the unequal group design in the LC2 structure, but it was still far from the true proportion (0.67). As for the LC3 structure, the mean of the classifier for the larger proportion in 1006s_lc3_e was 0.4 (SD = 0.04) and for the smaller proportion was 0.28 (SD = 0.03). The mean of the classifier for the larger proportion in 1006s_lc3_u was 0.4 (SD = 0.05) and for the smaller proportion was 0.28 (SD = 0.04).

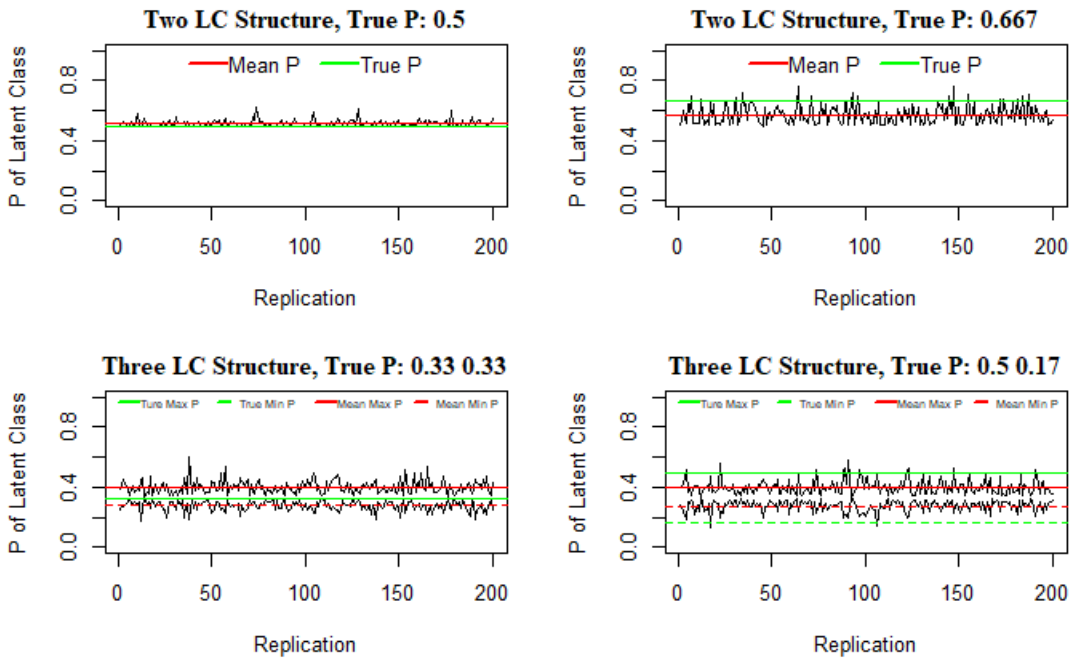

Figure 9
Classifier Parameter Recovery for Test 1006s

Classifier parameter recovery for test 1006g: the mean of the classifier for 1006s_lc2_e was 0.52 with a SD of 0.02 and the mean of classifier for 1006s_lc2_u was 0.54 with a SD of 0.04. The classifier parameter recovery showed some increase for the unequal group design in an LC2 structure, but it was still far from the true proportion (0.67). As for the LC3 structure, mean of the classifier for the larger proportion in 1006s_lc3_e was 0.4 (SD = 0.05) and for the smaller proportion was 0.27 (SD = 0.04). The mean of the classifier for the larger proportion in 1006s_lc3_u was 0.40 (SD = 0.05) and for the smaller proportion was 0.28 (SD = 0.04).

There was little difference in performance of classifier parameter recovery between 1006s and 1006g. The Rasch mixture model failed to distinguish LC size differences in the unequal design.

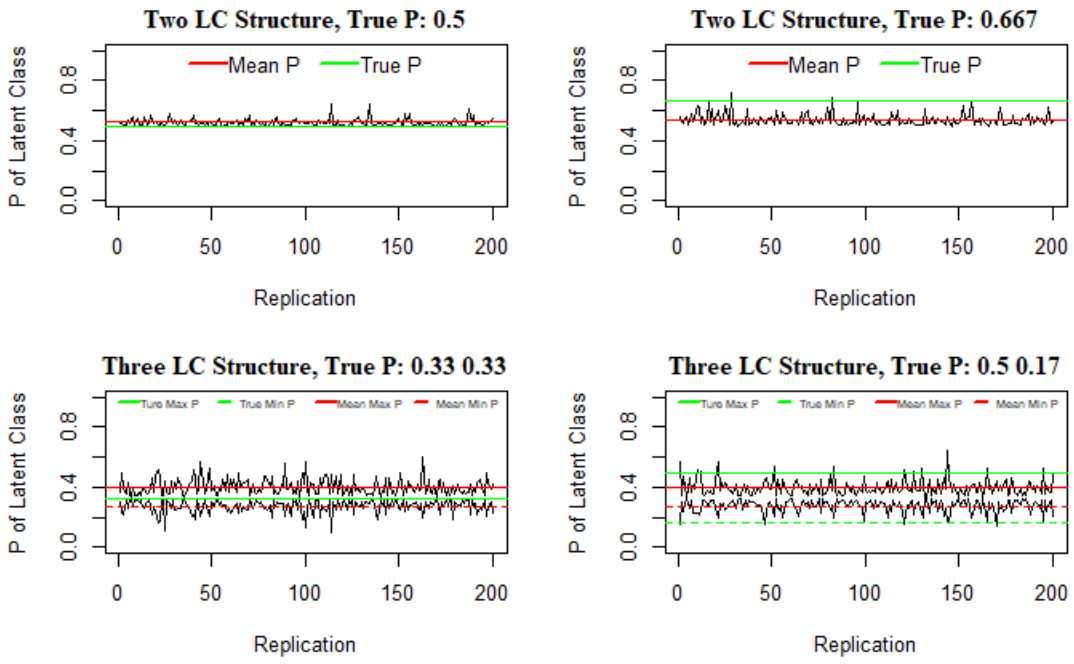

Figure 10
Classifier Parameter Recovery for Test 1006s

Classifier parameter recovery for test 3006s: the mean of the classifier parameter for 3006s_lc2_e was 0.52 with SD 0.01 and the mean of the classifier parameter for 3006s_lc2_u was 0.57 with SD 0.05. The mean of the classifier parameter for the largest proportion LC and smallest proportion LC in 3006s_lc3_e was 0.41 (SD = 0.05) and 0.26 (SD = 0.04), respectively. The mean of the classifier parameter for the largest proportion LC and smallest proportion LC in 3006s_lc3_u was 0.42 (SD = 0.06) and 0.26 (SD = 0.05).

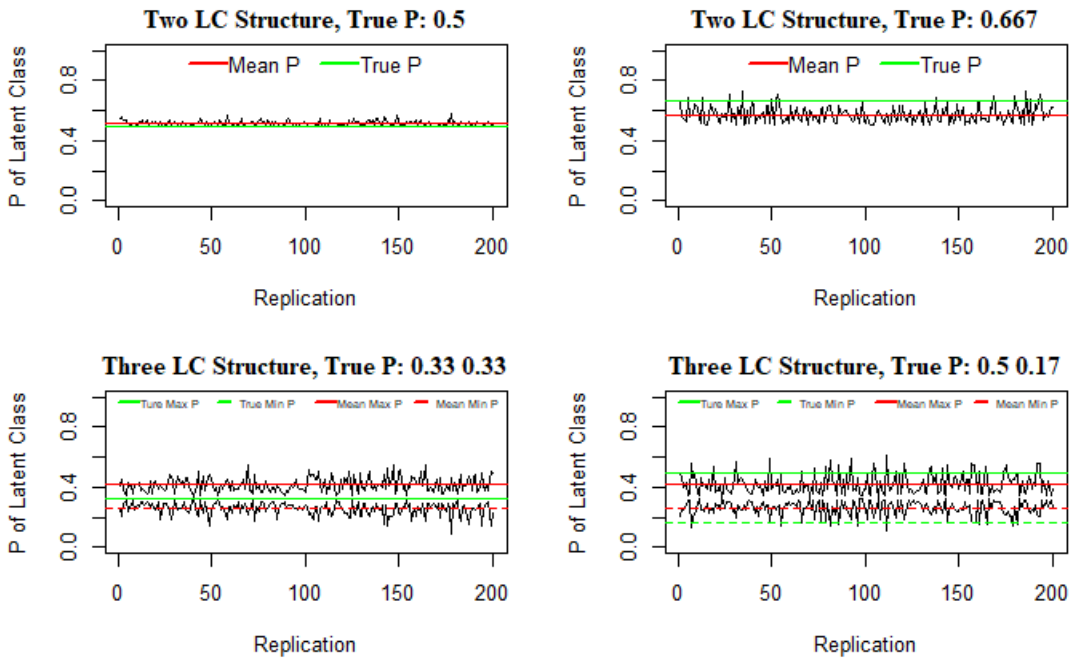

Figure 11
Classifier Parameter Recovery for Test 3006s

Classifier parameter recovery for test 3006g: the mean of the classifier parameter for 3006g_lc2_e was 0.52 with SD 0.02 and the mean of the classifier parameter for 3006g_lc2_u was 0.54 with SD 0.04. The mean of the classifier parameter for the largest proportion LC and smallest proportion LC in 3006g_lc3_e was 0.44 (SD = 0.05) and 0.24 (SD = 0.04) respectively. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3006s_lc3_u was 0.41 (SD = 0.06) and 0.27 (SD = 0.05).

Classifier parameter recoveries were acceptable for the dataset with equal LC sizes in both 3006s and 3006g. As tests with the same proportion (20%) of DIF items but larger test length, 3006s and 3006g had slightly better classifier parameter recovery than

tests with the same proportion (20%) of DIF items but shorter test length 1002s and 1002g.

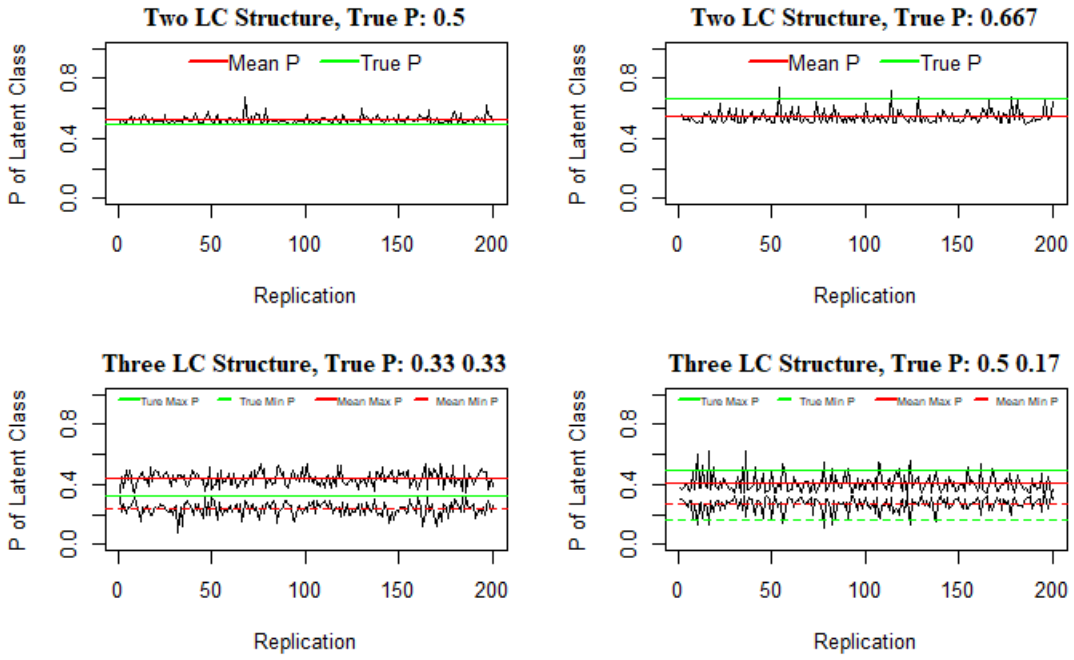

Figure 12
Classifier Parameter Recovery for Test 3006g

Classifier parameter recovery for test 3012s: the mean of the classifier parameter for 3012s_lc2_e was 0.52 with SD 0.01 and the mean of the classifier parameter for 3012s_lc2_u was 0.63 with SD 0.04. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3012s_lc3_e was 0.38 (SD = 0.03) and 0.29 (SD = 0.03), respectively. The mean of the classifier parameter for the largest proportion LC and smallest proportion LC in 3012s_lc3_u was 0.43 (SD = 0.05) and 0.25 (SD = 0.04).

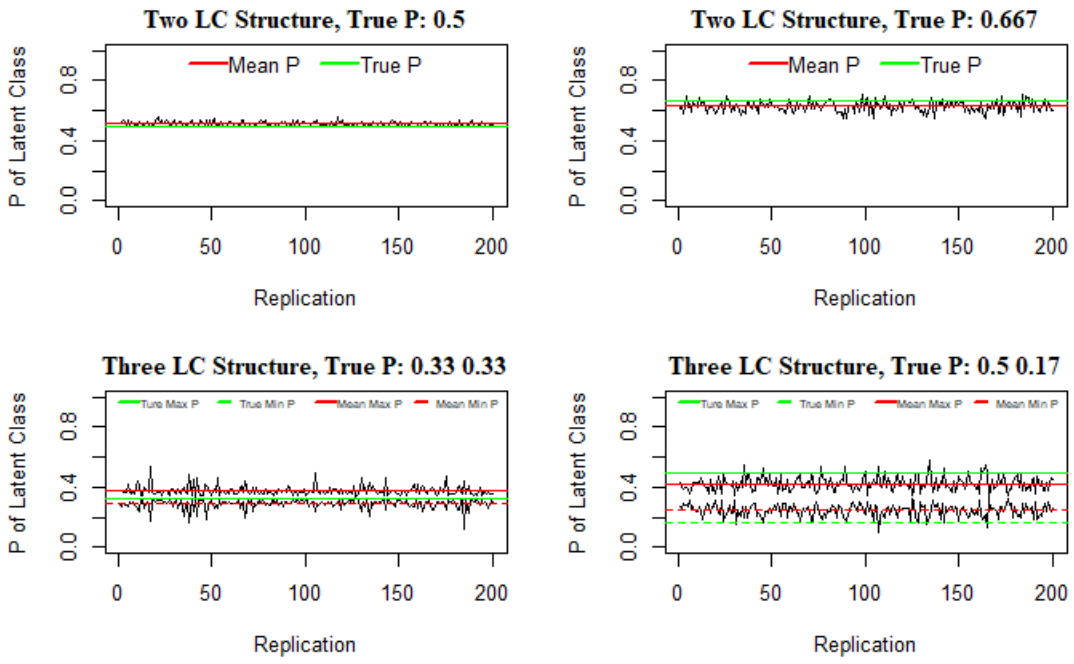

Figure 13
Classifier Parameter Recovery for Test 3012s

Classifier parameter recovery for test 3012g: the mean of the classifier parameter for 3012g_lc2_e was 0.52 with SD 0.02 and the mean of the classifier parameter for 3012g_lc2_u was 0.58 with SD 0.05. The mean of the classifier parameter for the largest proportion LC and smallest proportion LC in 3012g_lc3_e was 0.43 (SD = 0.05) and 0.25 (SD = 0.05), respectively. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3012g_lc3_u was 0.40 (SD = 0.06) and 0.27 (SD = 0.04).

As the mean of the simulated proportions got closer to true proportions, classifier parameter recoveries for test 3012 were better than those for test 3006. For test type 3012, the classifier recovery for the test with symmetric DIF pattern 3012s was better than that for the test with gradient DIF 3012g across all four LC size situations. Once

again, with the same proportion (40%) of DIF items, the test with longer test length (3012) showed better performance on classifier recovery than the test with shorter test length (1012).

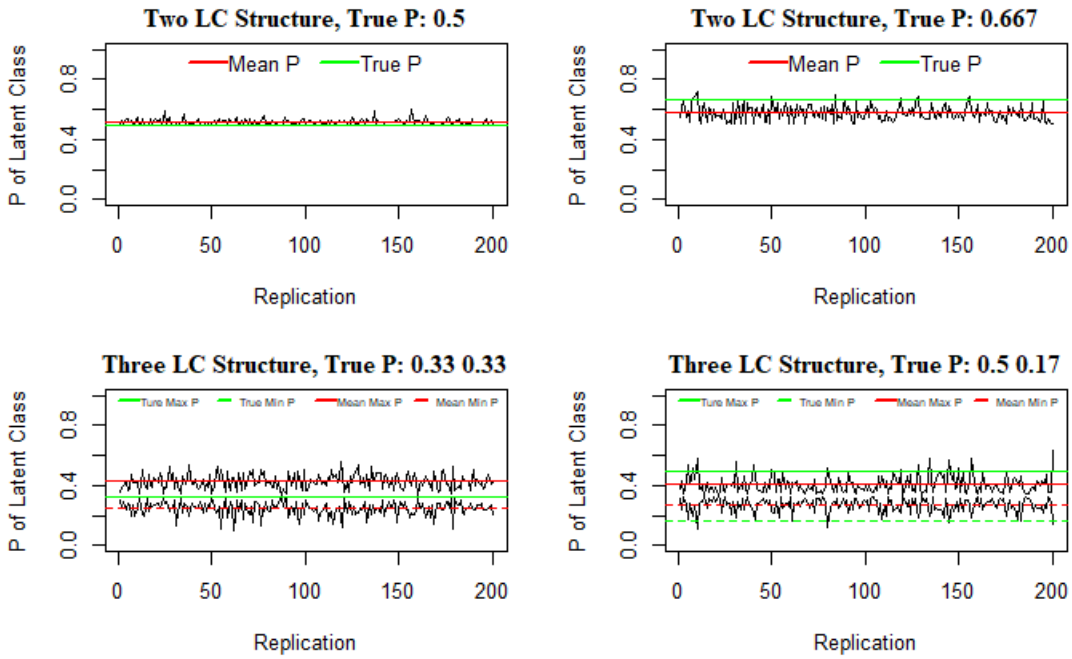

Figure 14
Classifier Parameter Recovery for Test 3012g

Classifier parameter recovery for test 3018s: the mean of the classifier parameter for 3018s_lc2_e was 0.51 with SD < 0.01 and the mean of the classifier parameter for 3018s_lc2_u was 0.65 with SD 0.02. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3018s_lc3_e was 0.36 (SD = 0.02) and 0.31 (SD = 0.02), respectively. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3018s_lc3_u was 0.46 (SD = 0.04) and 0.21 (SD = 0.04).

Classifier recovery for test 3018s was nearly perfect as all means of classifier parameters nearly overlapped with the true proportions, and there were small fluctuations across all four kinds of group size designs, which indicated stable performance of the Rasch mixture model.

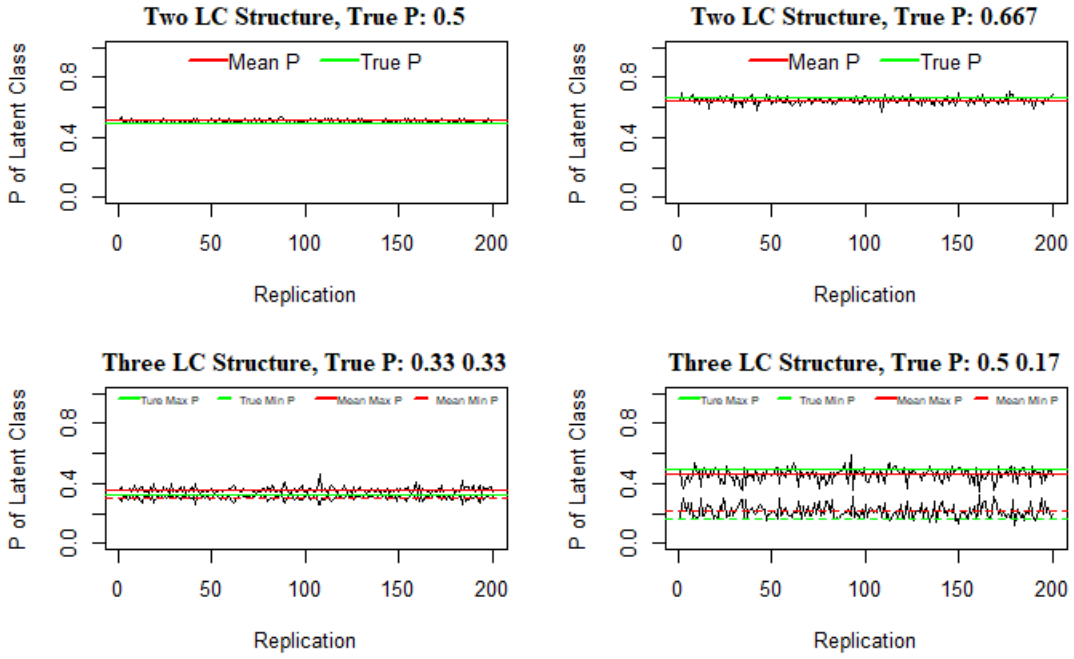

Figure 15
Classifier Parameter Recovery for Test 3018s

Classifier parameter recovery for test 3018g: The mean of the classifier parameter for 3018g_lc2_e was 0.52 with SD 0.01 and the mean of the classifier parameter for 3018g_lc2_u was 0.61 with SD 0.04. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3018g_lc3_e was 0.4 (SD = 0.04) and 0.28 (SD = 0.03), respectively. The mean of the classifier parameter for the largest proportion LC and the smallest proportion LC in 3018g_lc3_u was 0.41 (SD = 0.05) and 0.26 (SD = 0.04).

Classifier recovery was better for test for 3018s than that for test 3018g. For tests with 60% of DIF items, tests with longer length (3018s and 3018g) had better classifier recovery than tests with shorter length (1006s and 1006g).

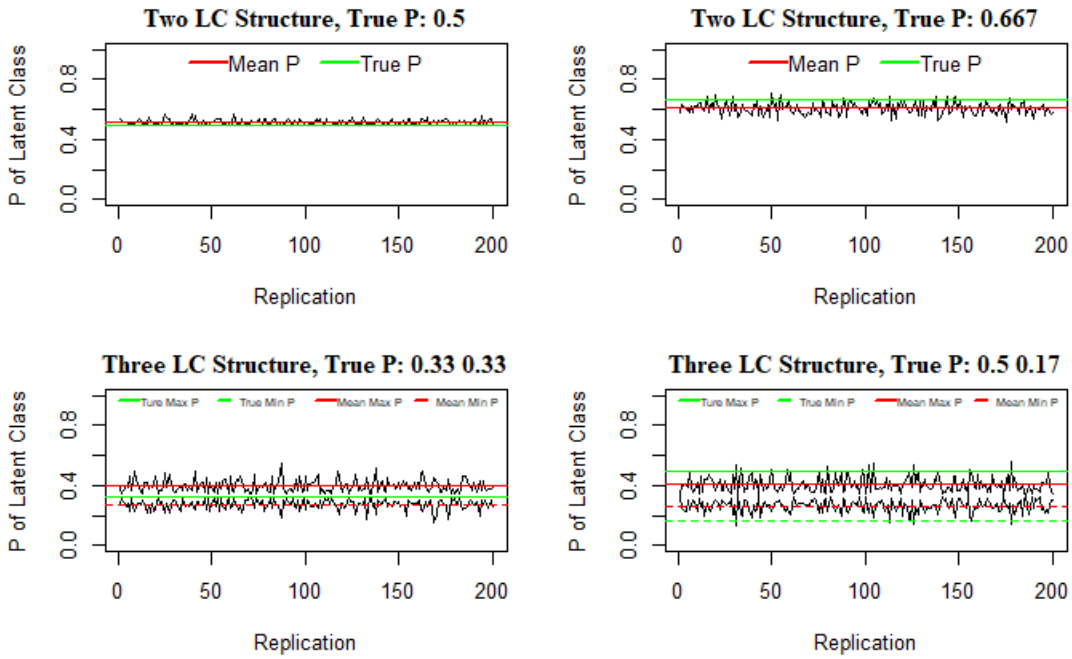

Figure 16
Classifier Parameter Recovery for Test 3018g

ANOVA on Classifier Parameter Recovery

An ANOVA was conducted to summarize effects of manipulated factors and interactions among them on classifier parameter recovery by using RMSE of classifier parameter as the dependent variable (Table 12). While the independence and normality assumptions of analysis of variance were met, a statistically significant ($p < .01$) violation of homogeneity was found for number of items, proportion of DIF, DIF type, LC structure and group size. However, analysis of variance is robust with respect to violation of homogeneity of variance with a balanced design.

Number of items was found to have a medium effect size, F (1,9552) = 943.88, η^2 = 0.09, with lower mean RMSE for 30-item tests (0.09, SD < 0.01) than for 10-item tests (0.12, SD < 0.01). Main effect of DIF type was found to have a small effect size, F (1,9552) = 544.10, η^2 = 0.05, with higher mean RMSE for tests with gradient DIF (0.11, SD < 0.01) than for tests with symmetric DIF (0.09, SD < 0.01). The main effect of LC structure was found to have a medium effect size, F (1,9552) = 908.58, η^2 = 0.09, with a higher mean RMSE for the three LC structure (0.12, SD < 0.01) than for the two LC structure (0.09, SD < 0.01). A medium effect size was found for the proportion of DIF, F (2,9552) = 417.80, η^2 = 0.08. Tukey's HSD post hoc test was used to examine differences for proportion of DIF. At the $p < 0.05$ level, significant differences were found between 20% DIF tests and 40% DIF tests, 20% DIF tests and 60% DIF tests, and 40% DIF tests and 60% DIF tests. Mean RMSE for 20% DIF tests, 40% DIF tests and 60% DIF tests were 0.12, 0.10 and 0.09, respectively. As proportion of DIF increased, RMSE decreased, which indicated an increase of RMM model fit from the perspective of classifier parameter recovery.

Table 12

Summary Table for Effects of Five Manipulated Factors on RMSE of Classifier Recovery

Source	Sum of Squares	df	Mean Square	F	*p*	η^2
n_of_items	2.090	1	2.090	943.879	< 0.001	.090
p_of_DIF	1.851	2	.925	417.795	< 0.001	.080
DIF_type	1.205	1	1.205	544.100	< 0.001	.054
LC_structure	2.012	1	2.012	908.576	< 0.001	.087
group_size	17.055	1	17.055	7700.865	< 0.001	.446
n_of_items * p_of_DIF	.577	2	.288	130.238	< 0.001	.027
n_of_items * DIF_type	.554	1	.554	250.099	< 0.001	.026
n_of_items * LC_structure	.501	1	.501	226.015	< 0.001	.023
n_of_items * group_size	2.263	1	2.263	1021.778	< 0.001	.097
p_of_DIF * DIF_type	.128	2	.064	28.971	< 0.001	.006
p_of_DIF * LC_structure	.085	2	.042	19.164	< 0.001	.004
p_of_DIF * group_size	.415	2	.207	93.593	< 0.001	.019
DIF_type * LC_structure	.001	1	.001	.362	.548	< 0.001
DIF_type * group_size	.179	1	.179	80.621	< 0.001	.008
LC_structure * group_size	3.033	1	3.033	1369.319	< 0.001	.125
n_of_items * p_of_DIF * DIF_type	.023	2	.011	5.096	.006	.001
n_of_items * p_of_DIF * LC_structure	.051	2	.025	11.445	< 0.001	.002
n_of_items * p_of_DIF * group_size	.006	2	.003	1.377	.252	< 0.001
n_of_items * DIF_type * LC_structure	.119	1	.119	53.516	< 0.001	.006
n_of_items * DIF_type * group_size	< 0.001	1	< 0.001	.071	.790	< 0.001
n_of_items * LC_structure * group_size	.171	1	.171	77.172	< 0.001	.008
p_of_DIF * DIF_type * LC_structure	.009	2	.005	2.061	.127	< 0.001
p_of_DIF * DIF_type * group_size	.044	2	.022	9.981	< 0.001	.002
p_of_DIF * LC_structure * group_size	.512	2	.256	115.697	< 0.001	.024

DIF_type * LC_structure * group_size	.269	1	.269	121.328	< 0.001	.013
n_of_items * p_of_DIF * DIF_type * LC_structure	.075	2	.037	16.865	< 0.001	.004
n_of_items * p_of_DIF * DIF_type * group_size	.014	2	.007	3.083	.046	.001
n_of_items * p_of_DIF * LC_structure * group_size	.253	2	.127	57.131	< 0.001	.012
n_of_items * DIF_type * LC_structure * group_size	.043	1	.043	19.370	< 0.001	.002
p_of_DIF * DIF_type * LC_structure * group_size	.039	2	.020	8.834	< 0.001	.002
n_of_items * p_of_DIF * DIF_type * LC_structure * group_size	.045	2	.023	10.253	< 0.001	.002
Error	21.154	9552	.002			
Total	156.149	9600				

There were six two-way interactions found to have interpretable effect sizes ($\eta^2 >$ 0.01) and they were: number of items by proportion of DIF, F (2,9552) = 130.24, η^2 = 0.03, number of items by DIF type, F (1,9552) = 250.10, η^2 = 0.03, number of items by LC structure, F (1,9552) = 226.02, η^2 = 0.02, number of items by group size, F (1,9552) = 1021.70, η^2 = 0.10, proportion of DIF by group size, F (2,9552) = 93.59, η^2 = 0.02, and LC structure by group size, F (1,9552), η^2 = 0.13. There were two three-way interpretable interactions which were proportion of DIF by LC structure by group size, F (2,9552) = 115.70, η^2 = 0.02, and DIF type by LC structure by group size, F (1,9553) = 121.33, η^2 = 0.01. There was one four-way interpretable interaction which was number of items by proportion of DIF by LC structure by group size, F (2,9552) = 57.13, η^2 = 0.01.

Figure 17 displays the group mean RMSEs for the number of items by proportion of DIF interaction. When controlling the number of items to be the same, tests with a larger proportion of DIF items had a lower mean RMSEs of classifier recovery. As the number of items increased from 10 items to 30 items, RMSE decreased for all levels of proportion of DIF but larger proportion of DIF had a larger RMSE decrease. Means and SDs of RMSE of classifier recovery for every level of number of items at each level of proportion of DIF are shown in Table 13.

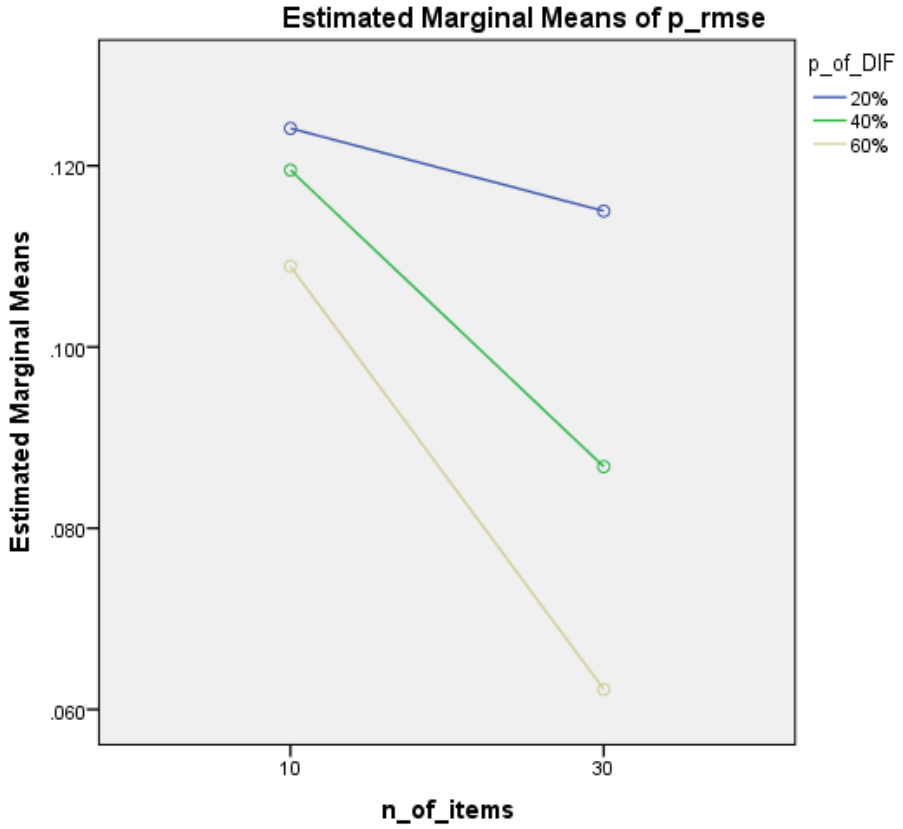

Figure 17
Plot for RMSE of Classifier Recovery for Number of Items by Proportion of DIF Interaction

Table 13
Means and SDs of RMSE of Classifier Recovery for Number of Items by Proportion of DIF Interaction

N of Items	P of DIF	Mean	SD
10	20%	.12	.001
	40%	.12	.001
	60%	.11	.001
30	20%	.12	.001
	40%	.09	.001
	60%	.06	.001

Figure 18 displays the group mean RMSEs for the number of items by DIF Type Interaction. When controlling the number of items, tests with a symmetric DIF pattern had a lower mean RMSEs of classifier recovery than tests with gradient DIF. As the number of items increased from 10 items to 30 items, RMSE decreased for both levels of DIF types but tests with symmetric DIF got larger RMSE decrease. Means and SDs of RMSE of classifier recovery for every level of number of items at each level of DIF type are shown in Table 14.

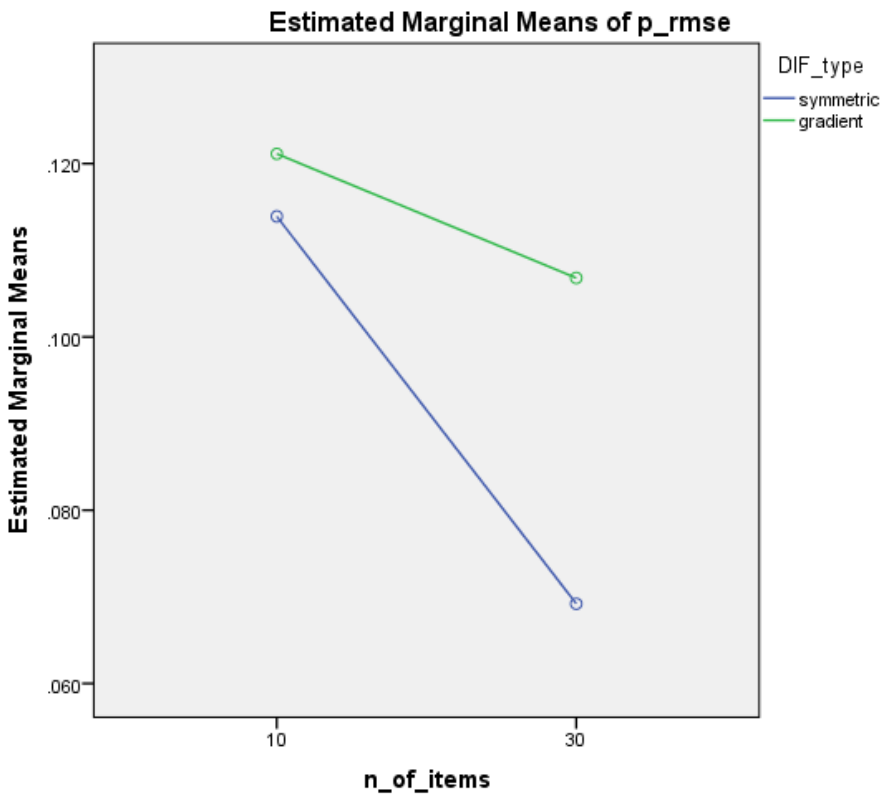

Figure 18
Plot for RMSE of Classifier Recovery for Number of Items by DIF Type Interaction

Table 14
Means and SDs of RMSE of Classifier Recovery for Number of Items by DIF Type Interaction

N of Items	DIF Type	Mean	SD
10	Symmetric	.11	.001
	Gradient	.12	.001
30	Symmetric	.07	.001
	Gradient	.11	.001

Figure 19 displays the group mean RMSEs for the number of items by LC structure interaction. When controlling the number of items, tests with the two LC structure had a lower mean RMSEs of classifier recovery than tests with the three LC structure. As the number of items increased from 10 items to 30 items, RMSE decreased for all levels of proportion of DIF but tests with a three LC structure showed a larger RMSE decrease. Means and SDs of RMSE of classifier recovery for every level of number of items at each level of LC structure are shown in Table 15.

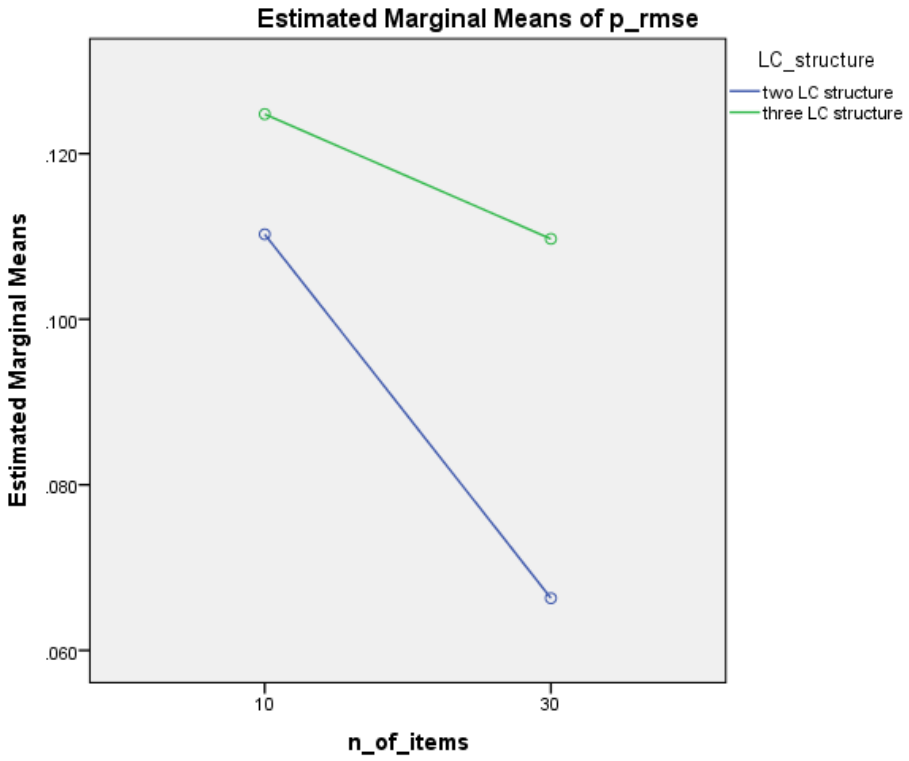

Figure 19
Plot for RMSE of Classifier Recovery for Number of Items by LC Structure Interaction

Table 15
Means and SDs of RMSE of Classifier Recovery for Number of Items by LC Structure Interaction

N of Items	LC Structure	Mean	SD
10	Two LC	.11	.001
	Three LC	.13	.001
30	Two LC	.07	.001
	Three LC	.11	.001

Figure 20 displays the group mean RMSEs for the number of items by group size interaction. When controlling the number of items, tests with an equal group size had a lower mean RMSE of classifier recovery than tests with an unequal group size. As the number of items increased from 10 items to 30 items, RMSE decreased for tests with an unequal group size but remained consistent for tests with two LC structure. Means and SDs of RMSE of classifier recovery for every level of number of items at each level of group size are shown in Table 16.

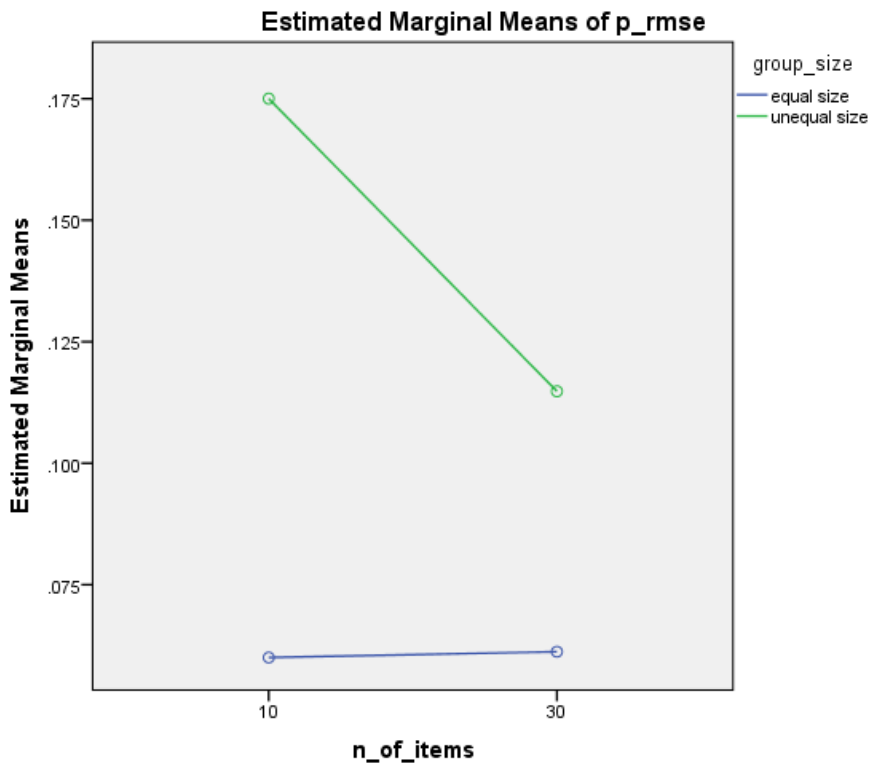

Figure 20
Plot for RMSE of Classifier Recovery for Number of Items by Group Size Interaction

Table 16
Means and SDs of RMSE of Classifier Recovery for Number of Items by Group Size Interaction

N of Items	Group Size	Mean	SD
10	Equal	.06	.001
	Unequal	.18	.001
30	Equal	.06	.001
	Unequal	.12	.001

Figure 21 displays the group mean RMSE for the proportion of DIF by group size interaction. When controlling the proportion of DIF, tests with an equal group size had a lower mean RMSEs of classifier recovery than tests with an unequal group size. As the proportion of DIF increased, RMSE decreased for both levels of group size but tests with the unequal group size had a larger RMSE decrease. Additionally, the trends of RMSE decreases for both levels of group size were consistent against the proportion of DIF. Means and SDs of RMSE of classifier recovery for every level of proportion of DIF at each level of group size are shown in Table 17.

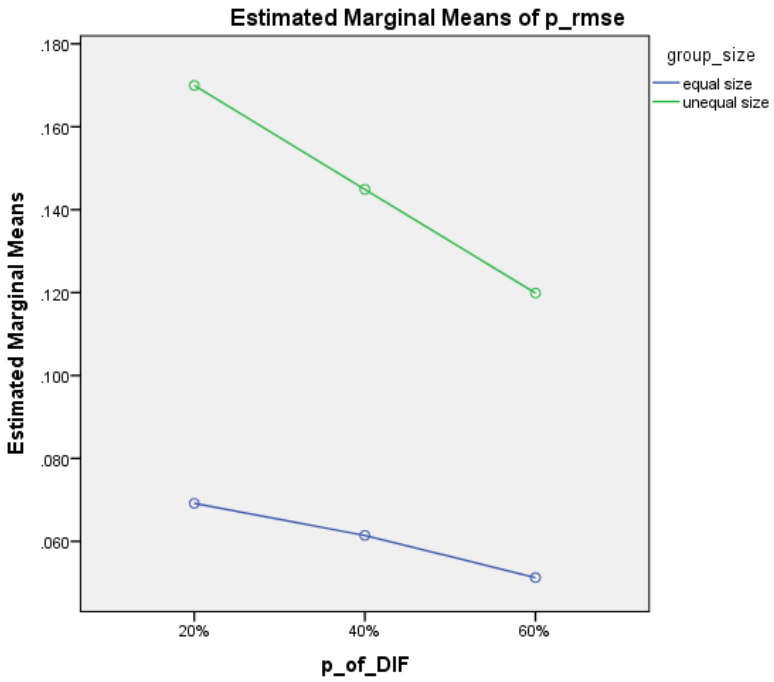

Figure 21
Plot for RMSE of Classifier Recovery for Proportion of DIF by Group Size Interaction

Table 17
Means and SDs of RMSE of Classifier Recovery for Proportion of DIF by Group Size Interaction

P of DIF	Group Size	Mean	SD
20%	Equal	.07	.001
	Unequal	.17	.001
40%	Equal	.06	.001
	Unequal	.15	.001
60%	Equal	.05	.001
	Unequal	.12	.001

Figure 22 displays the group mean RMSE for the LC structure by group size interaction. When controlling LC structure, tests with equal size had a lower mean RMSEs of classifier recovery than tests with unequal size. Means and SDs of RMSE of classifier recovery for every level of LC structure at each level of group size are shown in Table 18. The two LC structure with equal group size had a mean RMSE of 0.03 (SD < 0.01), the three LC structure with equal group size had a mean RMSE of 0.09 (SD < 0.01), the two LC structure with unequal group size had a mean RMSE of 0.15 (SD < 0.01), and the three LC structure with unequal group size had a mean RMSE of 0.14 (SD < 0.01). Means and SDs of RMSE of classifier recovery for every level of LC structure at each level of group size are shown in Table 18.

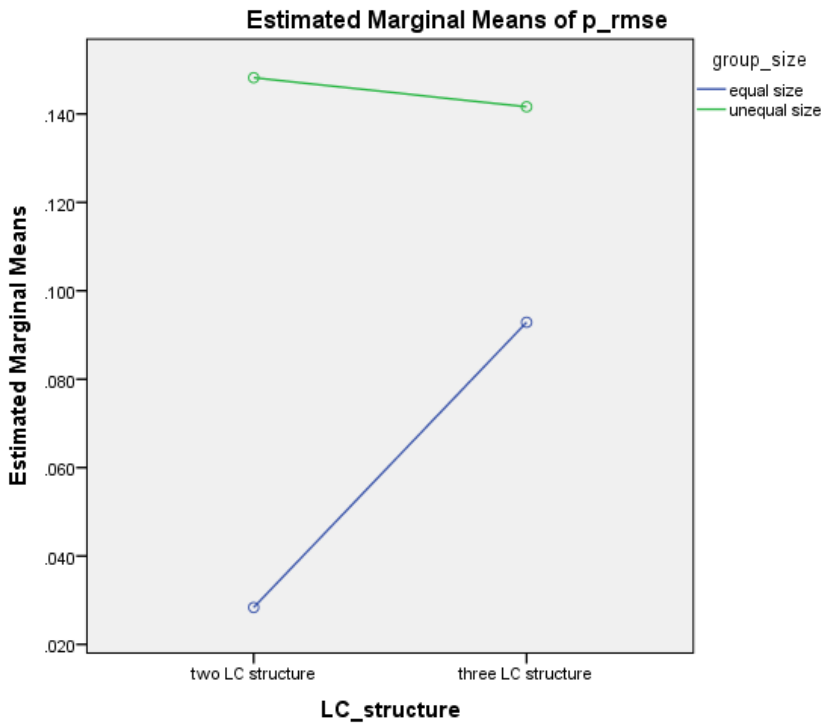

Figure 22
Plot for RMSE of Classifier Recovery for LC Structure by Group Size Interaction

Table 18
Means and SDs of RMSE of Classifier Recovery for LC Structure by Group Size Interaction

LC Structure	Group Size	Mean	SD
Two LC	Equal	.03	.001
	Unequal	.15	.001
Three LC	Equal	.09	.001
	Unequal	.14	.001

There was a small effect size (η^2 = 0.02) for the three-way interaction of proportion of DIF by LC structure by group size. Figure 23 shows this three-way interaction through splitting plots by group size. For equal group size tests, the interaction between LC structure and proportion of DIF was similar to the above two-way LC structure by proportion of DIF interaction. For unequal group size tests, at 20% DIF level, the two LC structure tests had larger RMSEs of (mean = 0.19, SD < 0.01) than the three LC structure tests (mean = 0.15, SD < 0.01); at 40% DIF level, the two LC structure tests had the same RMSEs of (mean = 0.15, SD < 0.01) as the three LC structure tests (mean = 0.15, SD < 0.01); at 60% DIF level, the two LC structure tests had lower RMSEs of (mean = 0.11, SD < 0.01) than the three LC structure tests (mean = 0.13, SD < 0.01). Means and SDs of RMSE of classifier recovery for every level of proportion of DIF at each level of LC structure at each level of group size are shown in Table 19.

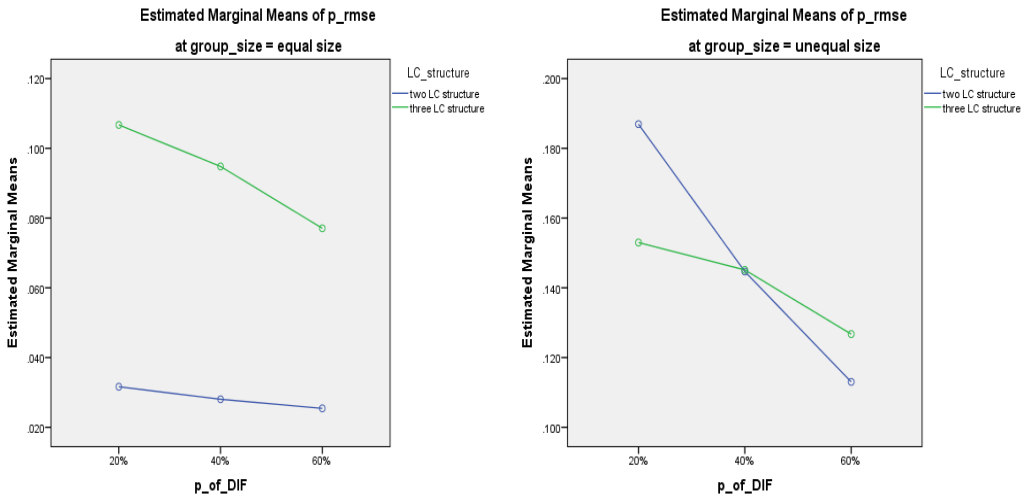

Figure 23
Plot for RMSE of Classifier Recovery for Proportion of DIF by LC Structure by Group Size Interaction

Table 19
Means and SDs of RMSE of Classifier Recovery for Proportion of DIF by LC Structure by Group Size Interaction

P of DIF	LC Structure	Group Size	Mean	SD
20%	Two LC	Equal	.03	.002
		Unequal	.19	.002
	Three LC	Equal	.11	.002
		Unequal	.15	.002
40%	Two LC	Equal	.03	.002
		Unequal	.14	.002
	Three LC	Equal	.10	.002
		Unequal	.15	.002
60%	Two LC	Equal	.03	.002
		Unequal	.11	.002
	Three LC	Equal	.08	.002
		Unequal	.13	.002

There was a small effect size ($\eta^2 = 0.01$) for the three-way interaction of DIF type by LC structure by group size. Figure 24 shows this three-way interaction through splitting plots by group size. For equal group size tests, the interaction between DIF type and proportion of DIF was similar to the above two-way DIF type by proportion of DIF interaction. For unequal group size, at the symmetric DIF level, the two LC structure tests had lower RMSEs of (mean = 0.13, SD < 0.01) than the three LC structure tests (mean = 0.13, SD < 0.01); at gradient DIF level, the two LC structure tests had larger RMSEs (mean = 0.17, SD < 0.01) than the three LC structure tests (mean = 0.15, SD < 0.01. Means and SDs of RMSE of classifier recovery for every level of DIF type at each level of LC structure at each level of group size are shown in Table 20.

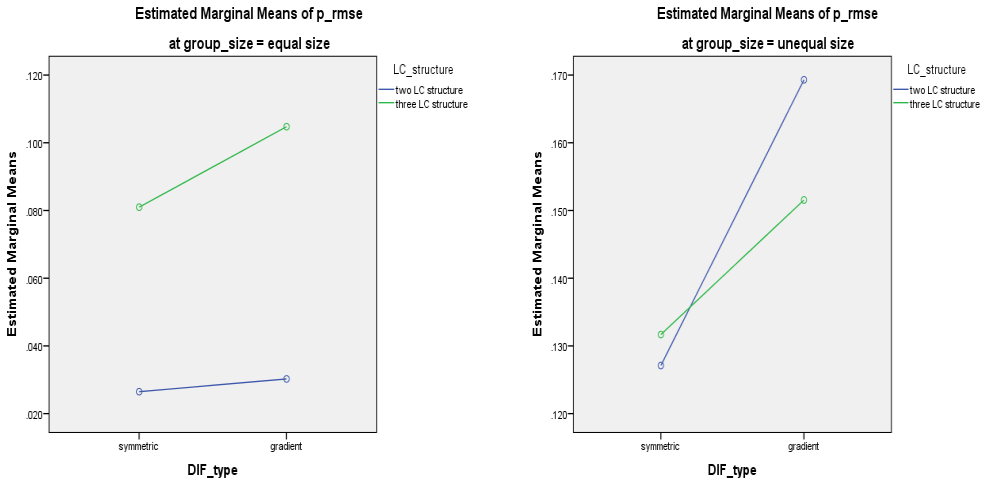

Figure 24
Plot for RMSE of Classifier Recovery for DIF Type by LC Structure by Group Size Interaction

Table 20
Means and SDs of RMSE of Classifier Recovery for DIF Type by LC Structure by Group Size Interaction

DIF Type	LC Structure	Group Size	Mean	SD
Symmetric	Two LC	Equal	.03	.001
		Unequal	.13	.001
	Three LC	Equal	.08	.001
		Unequal	.13	.001
Gradient	Two LC	Equal	.03	.001
		Unequal	.17	.001
	Three LC	Equal	.11	.001
		Unequal	.15	.001

There was a small effect size ($\eta^2 = 0.01$) for the only four-way interaction of number of items by DIF type by LC structure by group size interaction. Means and SDs

of RMSE of classifier recovery for every level of number of items at each level of DIF type at each level of LC structure at each level of group size are shown in Table 21.

Table 21
Means and SDs of RMSE of Classifier Recovery for Number of Items by Proportion of DIF by LC Structure by Group Size Interaction

N of Items	P of DIF	LC Structure	Group Size	Mean	SD
10	20%	Two LC	Equal	.03	.002
			Unequal	.21	.002
		Three LC	Equal	.08	.002
			Unequal	.17	.002
	40%	Two LC	Equal	.03	.002
			Unequal	.19	.002
		Three LC	Equal	.09	.002
			Unequal	.16	.002
	60%	Two LC	Equal	.03	.002
			Unequal	.16	.002
		Three LC	Equal	.09	.002
			Unequal	.15	.002
30	20%	Two LC	Equal	.03	.002
			Unequal	.16	.002
		Three LC	Equal	.13	.002
			Unequal	.14	.002
	40%	Two LC	Equal	.03	.002
			Unequal	.10	.002
		Three LC	Equal	.10	.002
			Unequal	.13	.002
	60%	Two LC	Equal	.02	.002
			Unequal	.06	.002
		Three LC	Equal	.06	.002
			Unequal	.10	.002

DIF Recovery

This section summarizes performance of the Rasch mixture model on detecting DIF from an overall perspective and individual item level across 48 simulation conditions. In order to avoid a label switching problem, DIF values generated from simulations were transformed into their absolute values. Correspondingly, when calculating MSE and RMSE, true DIF values were transformed into their absolute values.

For the two LC structure, DIF values for all items of a test including items with true DIF of 0.0 were calculated between two LC and then transformed to their absolute values. DIF recovery for the three LC structure had two parts: DIF (Δ_b) between the reference LC (θ = 0) which is $b_{LC_{f1}} - b_{LC_r}$ and LC with θ = 1, and DIF (2 Δ_b) between the reference LC and LC with θ = -1 which equals to $b_{LC_{f2}} - b_{LC_r}$. For instance, 1002s_lc3_e_1 refers to DIF between the reference LC and LC with θ = 1 for an equal size 10-item test with 20% of DIF in a symmetric pattern, and 3012g_lc3_u_2 refers to DIF between the reference LC and LC with θ = -1 for an unequal size designed 30-item test with 40% of DIF in a gradient pattern.

Overall Performance of the Rasch Mixture Model on Detecting DIF: Detailed MSE and RMSE values are shown in Table 22. A lower MSE and a lower RMSE suggest a better model fit. MSE and RMSE were larger for the three LC simulations than for the two LC simulations. However, it would be arbitrary to conclude that the Rasch mixture model had better DIF recovery for a two LC structure than for a three LC structure, because both MSE and RMSE took the number of items on the test into account. Both

MSE and RMSE were reported in this study, but only RMSE would be adequate in the future study.

Tests with DIF in a symmetric DIF pattern had lower MSE and RMSE than those with DIF in a gradient DIF pattern after controlling for other factors, which indicated a better DIF recovery of the symmetric pattern. For tests with the same test length, as the proportion of DIF increased, MSE and RMSE decreased indicating an increasing DIF recovery. When controlling other factors, MSE and RMSE were smaller for simulations with equal LC design than for those with unequal LC design.

Table 22
MSE and RMSE for 48 Simulated Conditions

Test Type	DIF	Index	Two LC		Three LC	
			Equal Size	*Unequal Size*	*Equal Size*	*Unequal Size*
1002	S	MSE	0.32	0.56	6.74	6.89
		RMSE	0.57	0.75	2.60	2.62
	G	MSE	2.17	2.76	9.73	8.75
		RMSE	1.47	1.66	3.12	2.96
1004	S	MSE	0.14	0.30	6.30	6.57
		RMSE	0.37	0.55	2.51	2.56
	G	MSE	2.47	2.83	14.86	15.18
		RMSE	1.57	1.68	3.85	3.90
1006	S	MSE	0.08	0.17	8.10	8.56
		RMSE	0.28	0.42	2.85	2.93
	G	MSE	4.95	5.29	25.88	25.52
		RMSE	2.22	2.30	5.09	5.05
3006	S	MSE	0.42	0.62	9.56	11.69
		RMSE	0.65	0.79	3.09	3.42
	G	MSE	1.99	2.65	16.56	16.80
		RMSE	1.41	1.63	4.07	4.10
3012	S	MSE	0.20	0.28	9.28	13.65
		RMSE	0.45	0.53	3.05	3.69
	G	MSE	3.93	4.13	26.83	26.23
		RMSE	1.98	2.03	5.18	5.12
3018	S	MSE	0.11	0.14	16.62	15.84
		RMSE	0.34	0.38	4.08	3.98
	G	MSE	8.34	8.46	47.93	48.27
		RMSE	2.89	2.91	6.92	6.95

Item Level DIF Recovery: Detailed item level DIF recovery logit mean differences are shown from Table 23 to Table 32 based on test type. In each of tables, M refers to mean of DIF values from 200 replications for the corresponding item. SD is the standard deviation of DIF values for a certain item. Detailed DIF recovery logit mean differences for each simulated scenario can be found in Appendix B.

DIF recovery logit mean differences for 10-item tests with 20% of DIF items (i.e., 1002) for s and g are shown in Table 23 and Table 24. For DIF items in 1002s, DIF recoveries were good for the two LC structure with equal group size and unequal group size except for item 2 DIF (M = 1.28, SD = 0.54) in 1002s_lc_u. But for the three LC structure, overall DIF recovery logit mean difference was not good except for 1002s_lc3_u_1 and all four types of three LC DIF recovery logit mean differences showed high standard deviations (SD > 1) except for item 1 (SD = 0.86) in 1002s_lc3_u_1.

For DIF items in 1002g, all DIF recoveries were poor except for 1002g_lc3_u_1. DIF recovery logit mean differences were less than half the size for items with true DIF = 2.00 in both two LC structures. For items with larger DIF (true DIF = 2.00), SD were bigger than those with smaller DIF (true DIF = 1.00). The Rasch mixture model failed to capture the true DIF between the reference LC ($\theta = 0$) and LC with $\theta = -1$ (1002g_lc3_e_2 and 1002g_lc3_u_2). All non-DIF items (true DIF = 0.00) had DIF recovery logit mean differences < 0.45.

Table 23

DIF Recovery Logit Mean Differences for 1002s

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.91	0.31	1.74	0.39	2.26	1.47	1.57	0.86	2.20	1.42	1.92	1.13
item 2	1.80	1.60	0.69	1.28	0.54	3.02	1.77	2.23	1.42	2.86	1.72	2.41	1.43
item 3	0.00	0.20	0.16	0.17	0.15	0.42	0.36	0.38	0.34	0.40	0.31	0.38	0.34
item 4	0.00	0.18	0.16	0.18	0.16	0.41	0.42	0.39	0.33	0.42	0.50	0.37	0.33
item 5	0.00	0.19	0.14	0.17	0.16	0.40	0.36	0.39	0.34	0.40	0.35	0.38	0.30
item 6	0.00	0.18	0.14	0.20	0.16	0.43	0.39	0.40	0.34	0.39	0.34	0.39	0.42
item 7	0.00	0.17	0.16	0.20	0.18	0.41	0.33	0.40	0.34	0.41	0.38	0.39	0.34
item 8	0.00	0.17	0.15	0.18	0.15	0.40	0.33	0.42	0.66	0.36	0.36	0.36	0.33
item 9	0.00	0.18	0.16	0.21	0.19	0.38	0.32	0.44	0.39	0.42	0.34	0.39	0.40
item 10	0.00	0.19	0.17	0.21	0.16	0.35	0.32	0.39	0.39	0.36	0.31	0.36	0.31

Table 24

DIF Recovery Logit Mean Differences for 1002g

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	2.00	0.96	0.78	0.73	0.56	3.02	1.78	2.21	1.43	2.67	1.89	2.29	1.60
item 2	1.00	0.82	0.73	0.68	0.55	1.20	0.65	1.12	0.83	1.20	0.73	1.16	0.85
item 3	0.00	0.38	0.24	0.36	0.26	0.67	0.45	0.53	0.34	0.61	0.52	0.56	0.37
item 4	0.00	0.35	0.26	0.37	0.26	0.60	0.38	0.57	0.37	0.63	0.40	0.56	0.37
item 5	0.00	0.35	0.26	0.37	0.30	0.61	0.37	0.52	0.34	0.59	0.39	0.55	0.35
item 6	0.00	0.38	0.28	0.37	0.36	0.63	0.50	0.54	0.34	0.61	0.40	0.55	0.36
item 7	0.00	0.37	0.30	0.38	0.26	0.61	0.39	0.58	0.40	0.58	0.37	0.57	0.40
item 8	0.00	0.36	0.26	0.35	0.25	0.69	0.41	0.56	0.39	0.67	0.53	0.57	0.39
item 9	0.00	0.34	0.27	0.36	0.28	0.60	0.37	0.58	0.38	0.61	0.39	0.60	0.37
item 10	0.00	0.37	0.28	0.33	0.27	0.64	0.42	0.57	0.43	0.63	0.40	0.60	0.43

DIF recovery logit differences for 10-item tests with 40% of DIF items (i.e., 1004) are shown in Table 25 and Table 26. DIF patterns (i.e., symmetric and gradient) were recovered for 12 situations of test 1004s. For 1004s, the two LC structures received good DIF parameter recoveries; parameter recovery had larger fluctuations for items with larger true DIF. 1004s_lc3_e_2 and 1004s_lc3_u_2 had mean DIF much less than desired DIF (2*True DIF). Similar to the two LC structure, SD of DIF got larger as DIF increased.

For 1004g, overall DIF recovery logit mean difference was lower than that in 1004s, particularly for the two LC structure in which DIF items only recovered about half of the true DIF. There were nearly no differences between 1004g_lc3_e_1, 1004g_lc3_e_2, 1004g_lc3_u_1 and 1004g_lc3_u_2. The Rasch mixture model failed to distinguish DIF for 1004g_lc3_e_2 and 1004g_lc3_u_2.

Table 25

DIF Recovery Logit Mean Differences for 1004s

		Two LC				Three LC							
		Equal Size		Unequal Size		Equal Size		Unequal Size		Equal Size		Unequal Size	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.80	0.22	1.73	0.33	2.40	1.41	1.87	0.98	2.32	1.52	1.94	0.93
item 2	-0.90	0.90	0.18	0.85	0.18	1.22	0.70	1.04	0.56	1.29	0.72	1.13	0.63
item 3	0.90	0.95	0.28	0.90	0.41	1.30	0.73	1.11	0.65	1.28	0.79	1.21	0.62
item 4	1.80	1.73	0.44	1.42	0.45	2.93	1.76	2.33	1.46	2.79	1.81	2.54	1.45
item 5	0.00	0.16	0.12	0.15	0.12	0.33	0.33	0.35	0.33	0.37	0.36	0.34	0.31
item 6	0.00	0.14	0.14	0.16	0.14	0.35	0.33	0.37	0.35	0.36	0.37	0.36	0.34
item 7	0.00	0.14	0.12	0.16	0.12	0.39	0.34	0.36	0.32	0.37	0.38	0.35	0.31
item 8	0.00	0.15	0.11	0.15	0.12	0.40	0.42	0.38	0.37	0.37	0.39	0.34	0.33
item 9	0.00	0.14	0.11	0.16	0.13	0.36	0.39	0.41	0.41	0.34	0.31	0.38	0.42
item 10	0.00	0.15	0.12	0.17	0.13	0.36	0.27	0.32	0.29	0.35	0.31	0.36	0.33

Table 26

DIF Recovery Logit Mean Differences for 1004g

		Two LC				Three LC							
		Equal Size		Unequal Size		Equal Size		Unequal Size		Equal Size		Unequal Size	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	2.00	1.38	0.55	1.07	0.50	2.45	1.77	2.15	1.45	2.45	1.45	2.16	1.47
item 2	1.50	1.02	0.42	0.85	0.47	1.58	1.10	1.42	0.87	1.49	1.13	1.43	1.02
item 3	1.00	0.55	0.28	0.46	0.28	0.78	0.49	0.74	0.44	0.89	0.77	0.77	0.49
item 4	0.50	0.17	0.16	0.20	0.26	0.41	0.38	0.41	0.41	0.41	0.40	0.45	0.60
item 5	0.00	0.48	0.16	0.45	0.24	0.86	0.60	0.78	0.46	0.77	0.46	0.80	0.47
item 6	0.00	0.51	0.19	0.42	0.19	0.88	0.54	0.75	0.38	0.84	0.52	0.78	0.46
item 7	0.00	0.52	0.18	0.45	0.20	0.80	0.45	0.81	0.46	0.82	0.49	0.77	0.47
item 8	0.00	0.53	0.18	0.45	0.19	0.81	0.44	0.80	0.46	0.82	0.49	0.78	0.47
item 9	0.00	0.50	0.16	0.44	0.20	0.83	0.47	0.77	0.43	0.83	0.50	0.81	0.50
item 10	0.00	0.50	0.20	0.43	0.20	0.85	0.52	0.78	0.52	0.88	0.54	0.79	0.47

DIF recovery logit mean differences for 10-item tests with 60% of DIF items (i.e., 1006) are shown in Table 27 and Table 28. Symmetric and gradient DIF patterns were tested across all 12 situations of test type 1006s. Standard deviation of recovered DIF increased as item true DIF increased. DIF recovery logit mean differences of 1006s_lc2_e and DIF recovery logit mean differences of 1006s_lc2_u almost overlapped with correspondingly true DIF. There was no difference on DIF recovery logit mean differences among 1006s_lc3_e_1, 1006s_lc3_e_2, 1006s_lc3_u_1 and 1006s_lc3_u_2.

However, RMM failed to recover the magnitude of true DIF for 1006g_lc2_e and 1006g_lc2_u. For the three LC structure of 1006g, there was no difference among 1006g_lc3_e_1, 1006g_lc3_e_2, 1006g_lc3_u_1 and 1006g_lc3_u_2. For test 1006 with a gradient DIF pattern, there was an increase of recovered DIF magnitude on non-DIF items compared to test 1006s, especially for 1006g_lc3 in which means of DIF for non-DIF items were greater than 1.

Table 27

DIF Recovery Logit Mean Differences for 1006s

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.82	0.2	1.68	0.24	2.52	1.29	2.07	1.08	2.29	1.47	1.95	1.05
item 2	-1.20	1.21	0.16	1.14	0.18	1.84	1.00	1.43	0.75	1.60	0.94	1.36	0.69
item 3	-0.60	0.59	0.17	0.56	0.16	1.00	0.57	0.88	0.59	0.85	0.53	0.81	0.44
item 4	0.60	0.63	0.23	0.6	0.22	0.88	0.48	0.74	0.44	0.88	0.48	0.72	0.42
item 5	1.20	1.22	0.29	1.10	0.26	1.87	0.82	1.57	0.93	1.63	0.80	1.43	0.83
item 6	1.80	1.74	0.41	1.55	0.47	2.99	1.52	2.53	1.53	2.59	1.41	2.34	1.38
item 7	0.00	0.13	0.11	0.14	0.11	0.33	0.34	0.38	0.36	0.34	0.35	0.37	0.36
item 8	0.00	0.13	0.11	0.14	0.12	0.32	0.34	0.30	0.30	0.35	0.37	0.33	0.32
item 9	0.00	0.14	0.13	0.14	0.12	0.31	0.53	0.34	0.32	0.31	0.31	0.32	0.30
item 10	0.00	0.14	0.12	0.15	0.12	0.34	0.37	0.33	0.29	0.32	0.32	0.38	0.46

Table 28

DIF Recovery Logit Mean Differences for 1006g

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	2.00	1.25	0.41	1.00	0.39	2.17	1.43	1.95	1.59	2.04	1.46	2.06	1.54
item 2	1.70	0.98	0.36	0.78	0.34	1.50	1.22	1.31	1.30	1.58	1.64	1.49	0.88
item 3	1.40	0.67	0.30	0.58	0.28	1.01	0.73	0.92	0.78	0.96	0.60	1.00	0.78
item 4	1.10	0.38	0.25	0.35	0.21	0.62	0.45	0.53	0.36	0.61	0.40	0.59	0.43
item 5	0.80	0.17	0.14	0.18	0.15	0.43	0.56	0.43	0.52	0.42	0.44	0.44	0.55
item 6	0.50	0.28	0.16	0.24	0.15	0.53	0.36	0.60	0.39	0.55	0.42	0.57	0.42
item 7	0.00	0.77	0.18	0.65	0.18	1.17	0.65	1.05	0.56	1.10	0.61	1.11	0.54
item 8	0.00	0.78	0.14	0.67	0.20	1.18	0.59	1.09	0.60	1.18	0.57	1.12	0.55
item 9	0.00	0.76	0.17	0.67	0.22	1.15	0.60	1.07	0.58	1.15	0.61	1.15	0.64
item 10	0.00	0.77	0.17	0.66	0.24	1.15	0.59	1.08	0.56	1.14	0.61	1.22	0.79

DIF recovery logit mean differences for 30-item tests with 20% of DIF items (i.e., 3006) are shown in Table 29 and Table 30. DIF patterns were restored (i.e., the recovered logit mean differences had same trends with true DIF patterns) for 12 DIF recovery situations for test type 3006. DIF magnitude for 3006s_lc2_e, 3006s_lc2_u, 3006s_lc3_e_1, 3006s_lc3_u_1, 3006g_lc3_e_1 and 3006g_lc3_u_1 was well captured. There was some shrinkage of recovered DIF for 3006g_lc2_e and 3006g_lc2_u. But 3006s_lc3_e_2, 3006s_lc3_u_2, 3006g_lc3_e_2 and 3006g_lc3_u_2 only achieved half of the true DIF.

Non-DIF items were well recovered with all means of recovered DIF below 0.30 for 3006s and below 0.50 for 3006g. There was only a small amount of variation of logit differences in non-DIF items as their SDs <= 0.35.

Table 29

DIF Recovery Logit Mean Differences for 3006s

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.86	0.17	1.74	0.20	2.22	1.21	1.83	0.85	2.06	1.15	1.72	0.83
item 2	-1.20	1.21	0.14	1.15	0.17	1.44	0.66	1.26	0.61	1.39	0.68	1.24	0.63
item 3	-0.60	0.58	0.13	0.58	0.15	0.85	0.43	0.75	0.40	0.78	0.41	0.72	0.38
item 4	0.60	0.63	0.21	0.56	0.20	0.85	0.42	0.73	0.41	0.79	0.40	0.74	0.41
item 5	1.20	1.19	0.27	1.09	0.30	1.68	0.75	1.68	0.85	1.66	0.79	1.51	0.85
item 6	1.80	1.68	0.34	1.49	0.44	2.87	2.08	2.62	1.67	2.81	1.66	2.43	1.62
item 7	0.00	0.13	0.12	0.15	0.11	0.24	0.22	0.26	0.21	0.25	0.22	0.27	0.43
item 8	0.00	0.12	0.09	0.14	0.11	0.24	0.21	0.23	0.20	0.24	0.21	0.26	0.23
item 9	0.00	0.12	0.10	0.14	0.12	0.27	0.26	0.25	0.24	0.27	0.24	0.24	0.22
item 10	0.00	0.13	0.10	0.14	0.10	0.26	0.23	0.29	0.26	0.24	0.20	0.26	0.22
item 11	0.00	0.14	0.11	0.15	0.11	0.22	0.19	0.27	0.23	0.23	0.22	0.25	0.23
item 12	0.00	0.13	0.09	0.16	0.12	0.26	0.21	0.25	0.19	0.25	0.20	0.27	0.19
item 13	0.00	0.14	0.11	0.15	0.11	0.25	0.22	0.26	0.26	0.24	0.24	0.30	0.28
item 14	0.00	0.13	0.11	0.16	0.12	0.24	0.27	0.24	0.20	0.25	0.24	0.28	0.25
item 15	0.00	0.13	0.10	0.15	0.12	0.22	0.19	0.25	0.23	0.27	0.24	0.24	0.20
item 16	0.00	0.13	0.09	0.14	0.10	0.22	0.18	0.27	0.24	0.23	0.19	0.27	0.23
item 17	0.00	0.13	0.11	0.14	0.12	0.25	0.20	0.25	0.18	0.24	0.22	0.28	0.22
item 18	0.00	0.12	0.09	0.15	0.12	0.26	0.24	0.26	0.22	0.24	0.20	0.27	0.24
item 19	0.00	0.13	0.11	0.13	0.11	0.25	0.21	0.25	0.24	0.26	0.23	0.26	0.25
item 20	0.00	0.13	0.11	0.13	0.12	0.28	0.35	0.25	0.27	0.27	0.31	0.27	0.26
item 21	0.00	0.13	0.11	0.14	0.11	0.25	0.21	0.26	0.22	0.25	0.20	0.27	0.20
item 22	0.00	0.13	0.11	0.16	0.12	0.24	0.25	0.26	0.20	0.24	0.22	0.25	0.21
item 23	0.00	0.13	0.11	0.16	0.15	0.26	0.23	0.24	0.25	0.25	0.21	0.28	0.24
item 24	0.00	0.13	0.10	0.14	0.11	0.25	0.26	0.25	0.21	0.25	0.24	0.26	0.22
item 25	0.00	0.13	0.11	0.14	0.11	0.22	0.17	0.28	0.25	0.24	0.19	0.26	0.20
item 26	0.00	0.14	0.11	0.15	0.12	0.24	0.21	0.23	0.20	0.25	0.24	0.26	0.23
item 27	0.00	0.13	0.10	0.14	0.12	0.25	0.25	0.28	0.26	0.24	0.26	0.29	0.29
item 28	0.00	0.13	0.10	0.14	0.13	0.23	0.21	0.25	0.19	0.24	0.23	0.23	0.21
item 29	0.00	0.13	0.09	0.14	0.12	0.25	0.23	0.28	0.25	0.26	0.25	0.28	0.27
item 30	0.00	0.12	0.10	0.14	0.11	0.22	0.20	0.27	0.23	0.24	0.20	0.25	0.23

Table 30

DIF Recovery Logit Mean Differences for 3006g

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f2}} - b_{LC_r}\|$		$\|b_{LC_{f2}} - b_{LC_r}\|$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	2.00	1.65	0.40	1.29	0.46	2.44	1.25	2.32	1.61	2.54	1.37	2.51	1.58
item 2	1.70	1.45	0.34	1.16	0.46	2.03	1.05	1.88	1.06	2.14	1.06	1.96	1.12
item 3	1.40	1.17	0.26	0.97	0.31	1.75	1.58	1.43	0.84	1.62	0.75	1.55	0.86
item 4	1.10	0.85	0.24	0.74	0.28	1.16	0.6	1.10	0.63	1.14	0.57	1.14	0.63
item 5	0.80	0.59	0.19	0.50	0.24	0.81	0.68	0.68	0.41	0.82	0.57	0.80	0.42
item 6	0.50	0.27	0.18	0.28	0.20	0.45	0.34	0.38	0.26	0.44	0.60	0.39	0.22
item 7	0.00	0.28	0.14	0.24	0.15	0.45	0.25	0.46	0.30	0.45	0.29	0.41	0.26
item 8	0.00	0.24	0.14	0.23	0.17	0.41	0.28	0.43	0.27	0.45	0.28	0.44	0.26
item 9	0.00	0.27	0.14	0.23	0.15	0.46	0.28	0.41	0.28	0.42	0.29	0.41	0.24
item 10	0.00	0.26	0.13	0.24	0.15	0.43	0.28	0.39	0.27	0.44	0.27	0.44	0.27
item 11	0.00	0.25	0.14	0.24	0.16	0.46	0.28	0.44	0.28	0.45	0.29	0.43	0.25
item 12	0.00	0.27	0.13	0.23	0.16	0.45	0.26	0.44	0.26	0.43	0.26	0.42	0.25
item 13	0.00	0.25	0.14	0.26	0.16	0.43	0.27	0.42	0.26	0.43	0.25	0.44	0.28
item 14	0.00	0.26	0.14	0.25	0.17	0.42	0.28	0.42	0.28	0.43	0.27	0.44	0.25
item 15	0.00	0.26	0.12	0.25	0.16	0.42	0.28	0.41	0.25	0.45	0.28	0.42	0.25
item 16	0.00	0.28	0.14	0.24	0.14	0.42	0.24	0.42	0.25	0.45	0.25	0.42	0.24
item 17	0.00	0.24	0.13	0.23	0.16	0.46	0.28	0.40	0.28	0.44	0.29	0.44	0.25
item 18	0.00	0.26	0.13	0.25	0.15	0.44	0.26	0.41	0.26	0.42	0.25	0.42	0.25
item 19	0.00	0.26	0.13	0.24	0.15	0.45	0.31	0.39	0.25	0.43	0.27	0.44	0.27
item 20	0.00	0.24	0.14	0.24	0.17	0.41	0.27	0.38	0.27	0.43	0.25	0.42	0.28
item 21	0.00	0.24	0.14	0.24	0.14	0.44	0.28	0.42	0.26	0.43	0.25	0.43	0.25
item 22	0.00	0.25	0.14	0.24	0.15	0.46	0.31	0.42	0.28	0.42	0.29	0.46	0.28
item 23	0.00	0.27	0.14	0.24	0.14	0.43	0.26	0.44	0.28	0.42	0.27	0.42	0.27
item 24	0.00	0.26	0.14	0.25	0.15	0.40	0.24	0.44	0.29	0.43	0.27	0.45	0.27
item 25	0.00	0.27	0.14	0.23	0.14	0.42	0.26	0.45	0.26	0.44	0.28	0.42	0.27
item 26	0.00	0.27	0.14	0.25	0.15	0.44	0.28	0.43	0.31	0.45	0.28	0.43	0.26
item 27	0.00	0.26	0.14	0.25	0.15	0.42	0.27	0.41	0.29	0.45	0.30	0.45	0.25
item 28	0.00	0.25	0.13	0.24	0.15	0.41	0.25	0.43	0.27	0.43	0.27	0.42	0.26
item 29	0.00	0.24	0.14	0.25	0.14	0.43	0.24	0.40	0.27	0.40	0.25	0.44	0.28
item 30	0.00	0.25	0.13	0.24	0.15	0.39	0.25	0.37	0.25	0.44	0.27	0.44	0.25

DIF recovery logit mean differences for 30-item tests with 40% of DIF items (i.e., 3012) are shown in Table 31 and Table 32. Both DIF patterns were well recovered for all 12 situations of test 3012. Variations for DIF recovery logit differences were low except for items (SD > 1.00) in the three LC structure with a more than medium level of DIF (DIF >= 1.50). 3012s_lc2_e and 3012s_lc2_u showed small DIF recovery logit mean differences but 3012g_lc2_e and 3012g_lc2_u demonstrated logit mean differences lower than true DIF. Once again, there was an increase in non-DIF items' DIF for 3012g compared to 3012s. DIF recovery logit mean differences for scenarios with equal group designs were better than those with unequal group designs. The Rasch mixture model still failed to distinguish true DIF between reference group ($\theta = 0$) and LC with $\theta = -1$.

Table 31

DIF Recovery Logit Mean Differences for 3012s

		Two LC				**Three LC**							
		Equal Size		**Unequal Size**		**Equal Size**		**Unequal Size**		**Equal Size**		**Unequal Size**	
		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f1}} - b_{LC_r}\|$		$\|b_{LC_{f2}} - b_{LC_r}\|$		$\|b_{LC_{f2}} - b_{LC_r}\|$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.80	0.13	1.76	0.15	2.33	1.21	2.09	1.19	2.44	1.21	1.96	1.23
item 2	-1.50	1.52	0.13	1.47	0.12	1.95	0.87	1.69	0.85	2.07	1.04	1.60	0.74
item 3	-1.20	1.21	0.13	1.18	0.13	1.61	0.86	1.38	0.64	1.62	0.66	1.39	0.80
item 4	-0.90	0.92	0.13	0.88	0.14	1.16	0.52	1.06	0.51	1.20	0.54	1.07	0.55
item 5	-0.60	0.59	0.12	0.58	0.13	0.84	0.40	0.75	0.39	0.84	0.39	0.73	0.40
item 6	-0.30	0.32	0.12	0.27	0.14	0.44	0.24	0.44	0.27	0.43	0.24	0.44	0.28
item 7	0.30	0.32	0.16	0.30	0.15	0.43	0.24	0.40	0.26	0.43	0.24	0.42	0.26
item 8	0.60	0.60	0.15	0.59	0.19	0.79	0.37	0.78	0.63	0.81	0.39	0.75	0.64
item 9	0.90	0.92	0.18	0.88	0.20	1.24	0.54	1.09	0.54	1.25	0.56	1.11	0.56
item 10	1.20	1.21	0.20	1.15	0.26	1.63	0.70	1.53	0.69	1.65	0.68	1.45	0.76
item 11	1.50	1.50	0.22	1.45	0.26	2.04	0.86	1.98	0.97	2.08	0.88	1.89	1.07
item 12	1.80	1.82	0.26	1.68	0.28	2.52	1.29	2.56	1.61	2.61	1.00	2.38	1.51
item 13	0.00	0.10	0.08	0.11	0.09	0.16	0.16	0.24	0.61	0.16	0.14	0.26	0.63
item 14	0.00	0.11	0.10	0.13	0.10	0.17	0.15	0.20	0.17	0.16	0.14	0.22	0.20
item 15	0.00	0.11	0.08	0.13	0.10	0.16	0.15	0.22	0.21	0.16	0.14	0.22	0.21
item 16	0.00	0.10	0.08	0.12	0.10	0.19	0.16	0.23	0.20	0.17	0.15	0.22	0.20
item 17	0.00	0.11	0.08	0.11	0.08	0.18	0.18	0.18	0.13	0.16	0.18	0.19	0.16
item 18	0.00	0.10	0.08	0.11	0.08	0.20	0.19	0.21	0.16	0.18	0.16	0.19	0.16
item 19	0.00	0.11	0.09	0.11	0.09	0.18	0.16	0.23	0.20	0.17	0.13	0.24	0.20
item 20	0.00	0.11	0.08	0.12	0.09	0.19	0.18	0.21	0.18	0.17	0.16	0.23	0.22
item 21	0.00	0.10	0.09	0.12	0.09	0.16	0.14	0.22	0.25	0.16	0.14	0.21	0.19
item 22	0.00	0.10	0.08	0.12	0.10	0.17	0.21	0.22	0.21	0.16	0.14	0.20	0.23
item 23	0.00	0.11	0.10	0.13	0.10	0.17	0.15	0.22	0.18	0.16	0.14	0.22	0.20
item 24	0.00	0.10	0.08	0.11	0.09	0.19	0.18	0.21	0.19	0.17	0.17	0.22	0.20
item 25	0.00	0.10	0.07	0.14	0.10	0.18	0.17	0.19	0.18	0.18	0.16	0.19	0.16
item 26	0.00	0.10	0.08	0.11	0.09	0.17	0.15	0.20	0.17	0.16	0.15	0.23	0.20
item 27	0.00	0.10	0.07	0.12	0.10	0.19	0.16	0.20	0.18	0.18	0.16	0.19	0.18
item 28	0.00	0.10	0.08	0.12	0.09	0.18	0.21	0.21	0.19	0.18	0.19	0.22	0.21
item 29	0.00	0.11	0.11	0.11	0.09	0.18	0.15	0.23	0.23	0.18	0.15	0.21	0.26
item 30	0.00	0.11	0.08	0.11	0.09	0.17	0.16	0.22	0.23	0.16	0.14	0.21	0.22

Table 32

DIF Recovery Logit Mean Differences for 3012g

		Two LC				Three LC							
		Equal Size		Unequal Size		Equal Size		Unequal Size		Equal Size		Unequal Size	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	2.00	1.57	0.28	1.35	0.35	2.35	1.14	2.16	1.26	2.32	1.31	2.32	1.23
item 2	1.80	1.41	0.30	1.24	0.35	2.11	0.97	1.77	0.98	2.00	1.21	2.03	1.17
item 3	1.60	1.25	0.34	1.11	0.31	1.77	0.97	1.55	0.81	1.72	0.88	1.67	0.84
item 4	1.40	1.06	0.23	0.92	0.25	1.43	0.63	1.28	0.68	1.37	0.68	1.38	0.66
item 5	1.20	0.84	0.20	0.79	0.24	1.19	0.54	1.02	0.57	1.13	0.51	1.12	0.55
item 6	1.00	0.61	0.18	0.60	0.22	0.85	0.41	0.78	0.47	0.88	0.43	0.87	0.42
item 7	0.80	0.42	0.18	0.40	0.20	0.64	0.33	0.55	0.30	0.64	0.35	0.59	0.33
item 8	0.60	0.22	0.15	0.23	0.16	0.37	0.25	0.35	0.24	0.39	0.24	0.36	0.22
item 9	0.40	0.12	0.10	0.15	0.30	0.21	0.20	0.22	0.20	0.22	0.21	0.21	0.17
item 10	0.30	0.12	0.09	0.14	0.11	0.25	0.24	0.27	0.20	0.25	0.21	0.28	0.18
item 11	0.20	0.19	0.12	0.18	0.12	0.35	0.21	0.39	0.26	0.36	0.24	0.37	0.22
item 12	0.10	0.26	0.13	0.26	0.13	0.47	0.29	0.42	0.25	0.47	0.30	0.45	0.25
item 13	0.00	0.39	0.13	0.36	0.16	0.58	0.29	0.53	0.29	0.54	0.31	0.57	0.30
item 14	0.00	0.37	0.13	0.35	0.14	0.55	0.28	0.59	0.47	0.57	0.30	0.58	0.33
item 15	0.00	0.38	0.13	0.33	0.14	0.59	0.29	0.57	0.30	0.55	0.31	0.55	0.29
item 16	0.00	0.39	0.14	0.36	0.15	0.60	0.31	0.53	0.31	0.57	0.30	0.57	0.32
item 17	0.00	0.37	0.15	0.33	0.16	0.55	0.30	0.51	0.29	0.57	0.31	0.56	0.27
item 18	0.00	0.37	0.14	0.36	0.15	0.57	0.31	0.54	0.30	0.56	0.30	0.56	0.29
item 19	0.00	0.39	0.13	0.37	0.14	0.57	0.26	0.53	0.33	0.54	0.29	0.55	0.30
item 20	0.00	0.39	0.12	0.36	0.16	0.55	0.29	0.51	0.30	0.55	0.28	0.58	0.28
item 21	0.00	0.37	0.14	0.35	0.15	0.57	0.27	0.55	0.31	0.55	0.30	0.56	0.29
item 22	0.00	0.38	0.12	0.36	0.14	0.59	0.31	0.54	0.30	0.57	0.31	0.58	0.29
item 23	0.00	0.40	0.15	0.33	0.15	0.61	0.32	0.57	0.33	0.56	0.33	0.56	0.37
item 24	0.00	0.39	0.15	0.33	0.15	0.57	0.29	0.52	0.31	0.56	0.29	0.58	0.30
item 25	0.00	0.39	0.15	0.37	0.14	0.59	0.30	0.54	0.30	0.57	0.31	0.59	0.31
item 26	0.00	0.39	0.13	0.34	0.14	0.58	0.37	0.51	0.31	0.57	0.33	0.58	0.30
item 27	0.00	0.37	0.14	0.36	0.16	0.59	0.29	0.52	0.31	0.57	0.32	0.57	0.33
item 28	0.00	0.40	0.14	0.32	0.15	0.59	0.29	0.54	0.33	0.57	0.38	0.56	0.31
item 29	0.00	0.38	0.13	0.35	0.16	0.60	0.31	0.52	0.32	0.53	0.29	0.60	0.29
item 30	0.00	0.38	0.13	0.33	0.15	0.60	0.29	0.54	0.30	0.56	0.29	0.56	0.30

DIF recovery logit mean differences for 30-item tests with 60% of DIF items (i.e., 3018) are shown in Table 33 and Table 34. DIF magnitudes were recovered well for the two LC structure, especially for 3018_lc2_e and 3018_lc2_u. DIF patterns were well restored except for the three LC structure with a gradient DIF pattern in which items (item 14 – item 18) with small true DIF (< 0.5) showed an opposite trend against the gradient pattern, and non-DIF items among these conditions received means of DIF around 0.76.

For the 30-item test, as proportion of DIF items increased from 20% to 60%, overall performance of the Rasch mixture model on DIF recovery logit mean differences decreased for the gradient DIF pattern and the means of DIF for non-DIF items increased.

Table 33

DIF Recovery Logit Mean Differences for 3018s

		Two LC				Three LC							
		Equal Size		Unequal Size		Equal Size		Unequal Size		Equal Size		Unequal Size	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	-1.80	1.82	0.12	1.77	0.13	2.60	1.23	2.30	1.41	2.36	1.28	2.19	0.96
item 2	-1.60	1.61	0.11	1.58	0.11	2.30	1.03	2.06	1.29	1.98	0.79	1.97	1.15
item 3	-1.40	1.40	0.11	1.40	0.12	1.93	0.78	1.89	1.33	1.74	0.75	1.66	0.83
item 4	-1.20	1.20	0.11	1.18	0.12	1.64	0.65	1.56	0.88	1.51	0.55	1.53	0.76
item 5	-1.00	1.01	0.11	0.98	0.13	1.42	0.61	1.29	0.61	1.28	0.49	1.24	0.59
item 6	-0.80	0.81	0.11	0.79	0.13	1.12	0.45	1.05	0.52	1.04	0.43	1.03	0.48
item 7	-0.60	0.60	0.11	0.59	0.13	0.83	0.36	0.85	0.44	0.76	0.35	0.81	0.41
item 8	-0.40	0.41	0.11	0.40	0.13	0.57	0.26	0.58	0.31	0.52	0.27	0.52	0.29
item 9	-0.20	0.19	0.11	0.20	0.12	0.31	0.18	0.33	0.21	0.26	0.18	0.31	0.24
item 10	0.20	0.21	0.12	0.21	0.12	0.30	0.18	0.29	0.19	0.29	0.18	0.28	0.20
item 11	0.40	0.39	0.13	0.39	0.14	0.59	0.26	0.53	0.28	0.49	0.27	0.50	0.27
item 12	0.60	0.60	0.14	0.59	0.14	0.84	0.35	0.77	0.38	0.77	0.34	0.73	0.33
item 13	0.80	0.79	0.15	0.79	0.18	1.15	0.43	1.05	0.49	1.01	0.41	1.00	0.42
item 14	1.00	1.02	0.16	1.01	0.19	1.43	0.52	1.29	0.60	1.28	0.52	1.27	0.50
item 15	1.20	1.22	0.16	1.19	0.21	1.79	1.31	1.59	0.70	1.64	1.48	1.54	0.64
item 16	1.40	1.43	0.19	1.38	0.20	1.98	0.73	1.88	0.81	1.76	0.69	1.83	0.75
item 17	1.60	1.60	0.18	1.59	0.28	2.30	0.84	2.22	1.18	2.06	0.78	2.07	0.84
item 18	1.80	1.80	0.22	1.75	0.28	2.63	0.97	2.48	1.12	2.31	1.09	2.30	1.01
item 19	0.00	0.10	0.08	0.10	0.08	0.15	0.12	0.18	0.15	0.15	0.14	0.18	0.15
item 20	0.00	0.09	0.07	0.11	0.08	0.13	0.12	0.18	0.16	0.15	0.12	0.17	0.15
item 21	0.00	0.10	0.08	0.11	0.10	0.13	0.11	0.19	0.17	0.15	0.12	0.18	0.16
item 22	0.00	0.09	0.07	0.10	0.08	0.15	0.13	0.17	0.17	0.15	0.14	0.17	0.16
item 23	0.00	0.10	0.08	0.10	0.08	0.14	0.14	0.18	0.17	0.15	0.14	0.17	0.17
item 24	0.00	0.09	0.07	0.11	0.09	0.15	0.15	0.18	0.16	0.15	0.15	0.16	0.14
item 25	0.00	0.09	0.07	0.11	0.09	0.13	0.11	0.23	0.25	0.14	0.12	0.20	0.17
item 26	0.00	0.09	0.07	0.11	0.09	0.15	0.13	0.18	0.15	0.14	0.14	0.15	0.13
item 27	0.00	0.11	0.08	0.11	0.10	0.15	0.13	0.20	0.18	0.15	0.14	0.19	0.16
item 28	0.00	0.09	0.07	0.11	0.08	0.14	0.15	0.16	0.14	0.15	0.16	0.15	0.13
item 29	0.00	0.09	0.07	0.10	0.08	0.15	0.14	0.18	0.15	0.17	0.15	0.18	0.17
item 30	0.00	0.10	0.08	0.10	0.08	0.15	0.13	0.18	0.21	0.16	0.14	0.18	0.18

Table 34

DIF Recovery Logit Mean Differences for 3018g

		Two LC				Three LC							
		Equal Size		Unequal Size		Equal Size		Unequal Size		Equal Size		Unequal Size	
		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f1}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$		$\lvert b_{LC_{f2}} - b_{LC_r} \rvert$	
	True DIF	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
item 1	1.80	1.21	0.24	1.12	0.30	1.68	0.74	1.62	0.86	1.67	0.72	1.75	0.92
item 2	1.70	1.14	0.27	1.04	0.25	1.54	0.68	1.49	0.79	1.57	0.98	1.54	0.77
item 3	1.60	1.00	0.22	0.94	0.27	1.37	0.60	1.32	0.66	1.41	0.62	1.40	0.70
item 4	1.50	0.93	0.21	0.88	0.30	1.25	0.61	1.20	0.62	1.32	0.65	1.22	0.64
item 5	1.40	0.84	0.21	0.76	0.23	1.13	0.66	1.08	0.57	1.12	0.53	1.06	0.53
item 6	1.30	0.74	0.20	0.70	0.21	0.99	0.50	0.95	0.50	0.98	0.47	0.96	0.51
item 7	1.20	0.63	0.19	0.61	0.20	0.90	0.69	0.79	0.46	0.86	0.42	0.85	0.43
item 8	1.10	0.55	0.17	0.53	0.19	0.71	0.37	0.70	0.47	0.75	0.38	0.72	0.43
item 9	1.00	0.44	0.18	0.39	0.18	0.59	0.35	0.58	0.32	0.57	0.31	0.58	0.30
item 10	0.90	0.35	0.16	0.33	0.19	0.48	0.27	0.48	0.27	0.49	0.27	0.46	0.27
item 11	0.80	0.24	0.13	0.25	0.17	0.38	0.22	0.32	0.20	0.37	0.24	0.35	0.24
item 12	0.70	0.17	0.12	0.18	0.15	0.27	0.22	0.29	0.25	0.26	0.20	0.26	0.20
item 13	0.60	0.12	0.09	0.14	0.12	0.20	0.18	0.25	0.22	0.22	0.19	0.24	0.25
item 14	0.50	0.12	0.10	0.15	0.11	0.25	0.21	0.24	0.22	0.24	0.19	0.25	0.20
item 15	0.40	0.18	0.12	0.18	0.12	0.30	0.22	0.36	0.28	0.31	0.21	0.37	0.29
item 16	0.30	0.27	0.14	0.26	0.14	0.41	0.23	0.45	0.28	0.39	0.24	0.45	0.25
item 17	0.20	0.37	0.13	0.35	0.14	0.52	0.31	0.54	0.26	0.54	0.28	0.53	0.32
item 18	0.10	0.48	0.14	0.44	0.15	0.62	0.31	0.64	0.34	0.65	0.31	0.67	0.35
item 19	0.00	0.58	0.12	0.56	0.14	0.80	0.39	0.75	0.39	0.77	0.39	0.76	0.37
item 20	0.00	0.58	0.13	0.54	0.14	0.80	0.37	0.75	0.40	0.76	0.36	0.76	0.39
item 21	0.00	0.56	0.13	0.55	0.15	0.76	0.39	0.76	0.34	0.76	0.36	0.76	0.38
item 22	0.00	0.57	0.15	0.54	0.14	0.78	0.40	0.73	0.39	0.78	0.37	0.76	0.38
item 23	0.00	0.58	0.13	0.54	0.14	0.78	0.36	0.77	0.36	0.79	0.38	0.78	0.40
item 24	0.00	0.56	0.14	0.53	0.14	0.76	0.36	0.76	0.38	0.76	0.36	0.77	0.36
item 25	0.00	0.57	0.14	0.54	0.14	0.75	0.36	0.75	0.38	0.81	0.38	0.76	0.39
item 26	0.00	0.57	0.13	0.54	0.17	0.78	0.36	0.76	0.37	0.78	0.40	0.79	0.38
item 27	0.00	0.57	0.13	0.54	0.16	0.76	0.36	0.77	0.40	0.80	0.37	0.79	0.41
item 28	0.00	0.57	0.13	0.54	0.16	0.76	0.38	0.74	0.40	0.78	0.35	0.78	0.36
item 29	0.00	0.59	0.13	0.53	0.15	0.76	0.36	0.76	0.36	0.75	0.37	0.74	0.36
item 30	0.00	0.57	0.13	0.54	0.15	0.76	0.36	0.78	0.39	0.79	0.4	0.73	0.41

ANOVA on DIF Recovery

An ANOVA was conducted to summarize effects of manipulated factors and interactions on DIF recovery by using RMSE as the dependent variable (Table 35). While the independence and normality assumptions of analysis of variance were met, a statistically significant ($p < .01$) violation of homogeneity was found for number of items, proportion of DIF, DIF type, LC structure, and group size. However, analysis of variance is robust with respect to violation of homogeneity of variance with a balanced design.

Number of items was found to have a medium-large effect size, F (1,9552) = 1887.92, $\eta^2 = 0.17$, with a higher mean RMSE for 30-item tests (4.17) than for 10-item tests (3.36). The main effect of DIF type was found to have a small effect size, F (1,9552) = 498.56, $\eta^2 = 0.05$, with higher mean RMSE for tests with gradient DIF (3.97) than for tests with symmetric DIF (3.55). The main effect of LC structure was found to have a large effect size, F (1,9552) = 56766.93, $\eta^2 = 0.86$, with a higher mean RMSE for the three LC structure (6.00) than for the two LC structure (1.52). A large effect size was found for proportion of DIF, F (2,9552) = 1208.02, $\eta^2 = 0.20$ Tukey's HSD post hoc test was used to examine differences for proportion of DIF. At the $p < 0.05$ level, significant differences were found between 20% DIF tests and 40% DIF tests, 20% DIF tests and 60% DIF tests, and 40% DIF tests and 60% DIF tests. Mean RMSE for 20% DIF tests, 40% DIF tests and 60% DIF tests were 3.25, 3.67 and 4.37, respectively.

Table 35

Summary Table for Effects of Five Manipulated Factors on RMSE of DIF Recovery

Source	Sum of Squares	df	Mean Square	F	*p*	η^2
n_of_items	1601.921	1	1601.921	1887.923	< .001	.165
p_of_DIF	2050.028	2	1025.014	1208.017	< .001	.202
DIF_type	423.031	1	423.031	498.558	< .001	.050
LC_structure	48167.289	1	48167.289	56766.933	< .001	.856
group_size	8.160	1	8.160	9.617	.002	.001
n_of_items * p_of_DIF	120.167	2	60.084	70.811	< .001	.015
n_of_items * DIF_type	1.157	1	1.157	1.364	.243	< .001
n_of_items * LC_structure	902.037	1	902.037	1063.084	< .001	.100
n_of_items * group_size	.134	1	.134	.158	.691	< .001
p_of_DIF * DIF_type	215.646	2	107.823	127.073	< .001	.026
p_of_DIF * LC_structure	1086.146	2	543.073	640.032	< .001	.118
p_of_DIF * group_size	.281	2	.140	.165	.848	< .001
DIF_type * LC_structure	2060.886	1	2060.886	2428.830	< .001	.203
DIF_type * group_size	.100	1	.100	.118	.731	< .001
LC_structure * group_size	8.160	1	8.160	9.617	.002	.001
n_of_items * p_of_DIF * DIF_type	1.503	2	.751	.886	.412	< .001
n_of_items * p_of_DIF * LC_structure	14.591	2	7.296	8.598	< .001	.002
n_of_items * p_of_DIF * group_size	.633	2	.316	.373	.689	< .001
n_of_items * DIF_type * LC_structure	50.315	1	50.315	59.298	< .001	.006
n_of_items * DIF_type * group_size	.061	1	.061	.071	.789	< .001
n_of_items * LC_structure * group_size	.134	1	.134	.158	.691	< .001
p_of_DIF * DIF_type * LC_structure	79.452	2	39.726	46.819	< .001	.010
p_of_DIF * DIF_type * group_size	.095	2	.048	.056	.945	< .001
p_of_DIF * LC_structure * group_size	.281	2	.140	.165	.848	< .001

DIF_type * LC_structure * group_size	.100	1	.100	.118	.731	< .001
n_of_items * p_of_DIF * DIF_type * LC_structure	39.133	2	19.567	23.060	< .001	.005
n_of_items * p_of_DIF * DIF_type * group_size	.067	2	.033	.039	.961	< .001
n_of_items * p_of_DIF * LC_structure * group_size	.633	2	.316	.373	.689	< .001
n_of_items * DIF_type * LC_structure * group_size	.061	1	.061	.071	.789	< .001
p_of_DIF * DIF_type * LC_structure * group_size	.095	2	.048	.056	.945	< .001
n_of_items * p_of_DIF * DIF_type * LC_structure * group_size	.067	2	.033	.039	.961	< .001
Error	8104.964	9552	.849			
Total	64937.326	9599				

There were five interactions found to have interpretable effect sizes ($\eta^2 > 0.01$) and they were: number of items by proportion of DIF (F (2,9552) = 70.81, $\eta^2 = 0.02$), number of items by LC structure (F (1,9552) = 1063.08, $\eta^2 = 0.10$), proportion of DIF by DIF type (F (2,9552) = 127.07, $\eta^2 = 0.03$), proportion of DIF by LC structure (F (2,9552) = 640.0, $\eta^2 = 0.12$), DIF type by LC structure (F (1,9552) = 2428.83, $\eta^2 = 0.20$).

Tests with a higher proportion of DIF items had higher mean RMSE for both the gradient DIF pattern and the symmetric pattern (Figure 25). As the test length increased from 10 items to 30 items, mean RMSE increased for all three levels of proportion of DIF variable and tests with a larger proportion of DIF had a higher mean RMSE increase. Mean RMSEs and their SDs for every level of number of items at each level proportion of DIF are shown in Table 36.

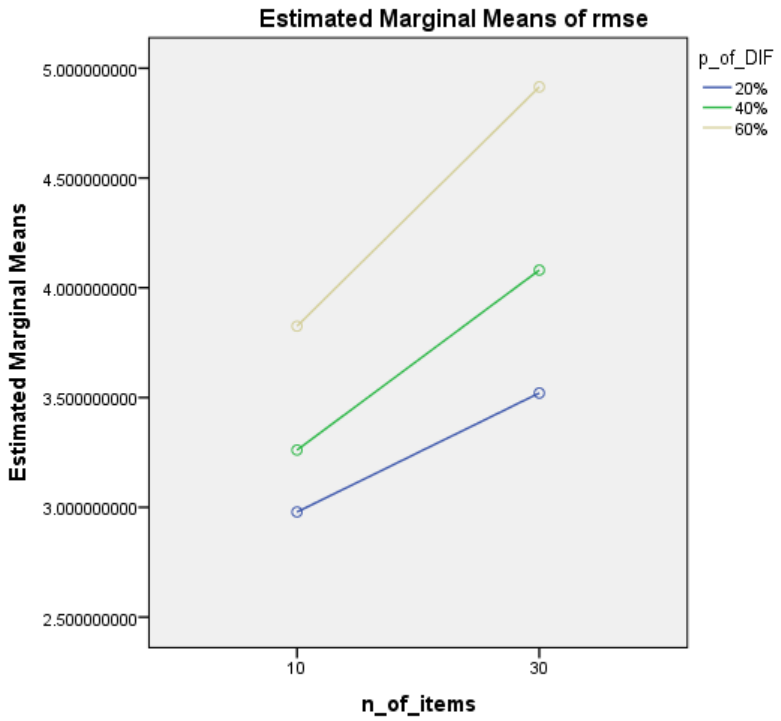

Figure 25
Plot for RMSE of DIF Recovery for Proportion of DIF by Number of Items Interaction

Table 36
Means and SDs of RMSE of DIF Recovery for Proportion of DIF by Number of Items Interaction

N of Items	P of DIF	Mean	SD
10	20%	2.98	.023
	40%	3.26	.023
	60%	3.83	.023
30	20%	3.52	.023
	40%	4.08	.023
	60%	4.92	.023

Tests with the three LC structure had a higher mean RMSE compared to those with the two LC structure for both levels of number of items. Figure 26 displays of

number of items by LC structure interaction. As number of items increased from 10 items to 30 items, both mean RMSEs of two levels of LC structure variable increased and the three LC structure level showed larger increased mean RMSE. Means and SDs of RMSE for every level of number of items at each level of LC structure are shown in Table 37.

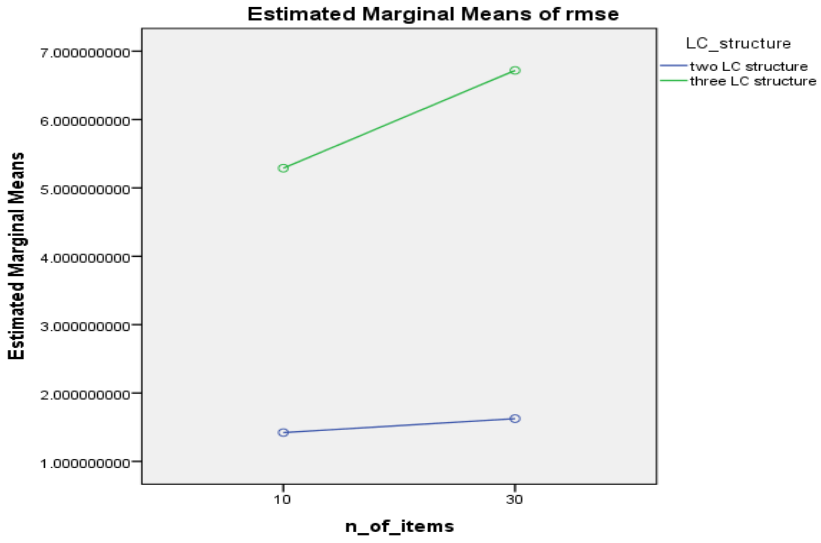

Figure 26
Plot for RMSE of DIF Recovery for Number of Items by LC Structure Interaction

Table 37
Means and SDs of RMSE of DIF Recovery for Number of Items by LC Structure Interaction

N of Items	LC Structure	Mean	SD
10	Two LC	1.42	.019
	Three LC	5.29	.019
30	Two LC	1.63	.019
	Three LC	6.72	.019

Tests with a symmetric DIF pattern had higher mean RMSEs for every level of proportion of DIF (Figure 27). As the proportion of DIF increased, the mean RMSE increased for both the symmetric level and the gradient level of DIF type. Specifically, the increased mean RMSE for symmetric level was approximately equal between from 20% to 40% and from 40% to 60%, but the increased mean RMSE for the gradient level was larger from 40% to 60% than from 20% to 40%. Means and SDs of RMSE for every level of proportion of DIF at each level of DIF type are shown in Table 38.

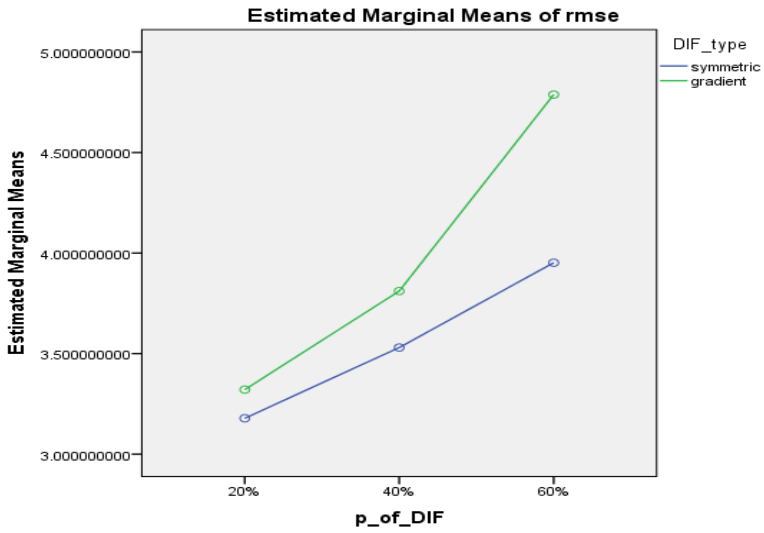

Figure 27
Plot for RMSE of DIF Recovery for DIF Type by Proportion of DIF Interaction

Table 38
Means and SDs of RMSE of DIF Recovery for DIF Type by Proportion of DIF Interaction

P of DIF	DIF Type	Mean	SD
20%	Symmetric	3.18	.023
	Gradient	3.32	.023
40%	Symmetric	3.53	.023
	Gradient	3.81	.023
60%	Symmetric	3.95	.023
	Gradient	4.79	.023

Tests with a three LC structure had higher mean RMSEs than those with a three LC structure (Figure 28). At the two LC structure level, mean RMSE increased as the proportion of DIF increased. In contrast, at the three LC structure level, mean RMSE remained relatively consistent as the proportion of DIF increased from 20% to 60%.

Means and SDs of RMSE for every level of proportion of DIF at each level of DIF structure are shown in Table 39.

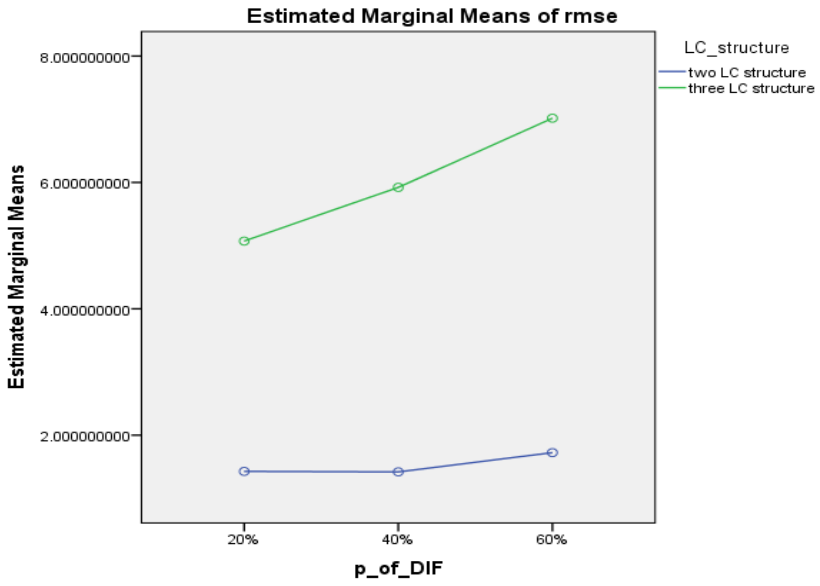

Figure 28
Plot for RMSE of DIF Recovery for Proportion of DIF by LC Structure Interaction

Table 39
Means and SDs of RMSE of DIF Recovery for Proportion of DIF by LC Structure Interaction

P of DIF	LC Structure	Mean	SD
20%	Two LC	1.43	.023
	Three LC	5.07	.023
40%	Two LC	1.42	.023
	Three LC	5.92	.023
60%	Two LC	1.72	.023
	Three LC	7.02	.023

Tests with a three LC structure had larger mean RMSEs than those with a two LC structure (Figure 29). Symmetric tests with two LC structure had the lowest mean RMSE = 0.85 (SD = 0.02). Gradient tests with a two LC structure had mean RMSE = 2.20 (SD = 0.02). Symmetric tests with a three LC structure had mean RMSE = 6.26 (SD = 0.02). Gradient tests with a three LC structure had mean RMSE = 5.75 (SD = 0.02). Means and SDs of RMSE for every level of DIF type at each level of LC structure are shown in Table 40.

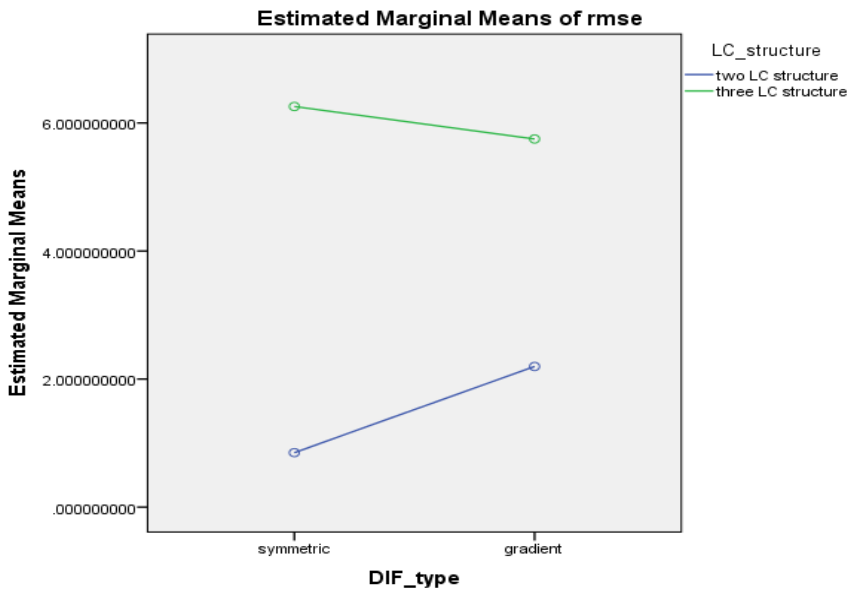

Figure 29
Plot for RMSE of DIF Recovery for DIF type by LC Structure Interaction

Table 40
Means and SDs of RMSE of DIF Recovery for DIF type by LC Structure Interaction

DIF Type	LC Structure	Mean	SD
Symmetric	Two LC	.85	.019
	Three LC	6.26	.019
Gradient	Two LC	2.20	.019
	Three LC	5.75	.019

Simulation Running Time for Each Scenario

Total running time for 200 replications on parameter recovery of each scenario was recorded (Table 41). There were 101 hours computation time spent for the whole simulation process in which 75 hours were for classifier parameter recovery and 26 hours were for DIF recovery. Simulations ran in a PC with Intel Core i7-7700HQ CPU @ 2.8GHz with 16 GB RAM.

30-item tests which had more computational complexity due to more items had longer running time than 10-item tests. Tests with a symmetric DIF pattern had shorter running times compared to corresponding tests with a gradient DIF. Tests dealing with a three LC structure had longer running times than their two LC structure counterparts. For 10-item tests, there was nearly no difference in running time between tests with an equal group design and tests with an unequal design. 30-item tests with the three LC structure were the most time consuming and each of them cost about an hour.

Table 41
Running Time for Each Scenario on DIF Recovery in Second

Test Type	**DIF Pattern**	**Two LC**		**Three LC**	
		Equal Size	*Unequal Size*	*Equal Size*	*Unequal Size*
1002	S	464.14	442.83	1270.36	1147.28
	G	641.39	631.11	1081.89	1144.39
1004	S	365.01	474.41	1048.95	1067.76
	G	555.63	570.23	1047.03	1077.50
1006	S	293.38	503.09	1041.81	1032.85
	G	465.27	515.36	1073.80	1061.47
3006	S	1086.67	1622.21	3319.58	3235.00
	G	1575.59	1978.75	3618.64	3242.73
3012	S	811.49	1509.25	2781.60	3357.55
	G	1211.56	1903.32	3488.00	3345.07
3018	S	654.92	1215.33	2234.82	3855.42
	G	1054.73	1833.37	3385.49	3753.18

Summary of Results

Two ANOVAs were conducted to assess effects of five manipulated factors on latent structure recovery. One of them used *ln* (AIC) as the dependent variable and the other one used *ln* (BIC) as the dependent variable. The natural log was used to normalize AIC and BIC. Lower *ln* (AIC) and lower *ln* (BIC) indicate better latent structure recovery. The results for the two ANOVAs were very similar. There were interpretable main effect sizes ($\eta^2 > 0.01$) for all five manipulated factors: 10-items tests had better latent structure recovery than 30-item tests; as the proportion of DIF increased, latent structure recovery got better; tests with gradient DIF had better latent structure recovery than those with symmetric DIF; three LC structure tests had better latent structure recovery than two LC structure; tests with equal group size showed better latent structure recovery than those with unequal group size.

There was one similar interpretable interaction for both latent structure recovery ANOVAs, which was LC structure by DIF type. At the symmetric level of DIF type, three LC structure tests showed better latent structure recovery, but when it came to gradient level of DIF type, three LC structure tests and two LC structure tests showed similar latent structure recovery.

Two ANOVAs were conducted to examine the effects of manipulated factors on parameter recovery. One ANOVA was for classifier parameter recovery using RMSE as the dependent variable which was calculated from predicted proportion of LC and true proportion of LC. The other ANOVA was for DIF recovery using RMSE as the dependent variable and the RMSE was calculated based on predicted DIF and true DIF.

Lower RMSE indicates better parameter prediction (i.e., parameter recovery). All five manipulated factors showed interpretable main effect size ($\eta^2 > 0.01$) on classifier recovery. However, for DIF recovery, all factors except group size showed interpretable main effect sizes ($\eta^2 > 0.01$). The directions of interpretable main effects varied for the number of items factor and the proportion of DIF: for the former, as number of items increased from 10 items to 30 items, RMSE of classifier recovery decreased, but for DIF recovery, as number of items increased from 10 items to 30 items RMSE of DIF recovery increased; for the latter, as proportion of DIF items increased RMSE of classifier recovery decreased, but as proportion of DIF items increased RMSE of DIF recovery increased. Directions of main effects of DIF type and LC structure were consistent from classifier recovery to DIF recovery: tests with symmetric DIF showed lower RMSE than tests with gradient DIF, two LC structure tests showed lower RMSE than three LC structure. Group size showed a strong effect size ($\eta^2 = 0.45$) on classifier recovery with lower RMSE for equal group size tests than unequal group size tests, but there was no interpretable effect size of group size on DIF recovery.

There were nine interactions for classifier recovery: six two-way interactions, two three-way interactions and one four-way interaction. Seven of the interpretable interactions for classifier showed a small effect size ($0.01 < \eta^2 \leq 0.06$). Two interpretable interactions for DIF recovery showed small-medium effect sizes ($0.06 < \eta^2 \leq 0.14$) and they were number of items by group size interaction ($\eta^2 = 0.10$) and LC structure by group size ($\eta^2 = 0.13$).

There were five interpretable interactions for DIF recovery and all of them were two-way interactions. Two interpretable interactions for DIF recovery had small effect sizes ($0.01 < \eta^2 \leq 0.06$) and they were number of items by proportion of DIF interaction ($\eta^2 = 0.12$) and proportion of DIF by DIF type ($\eta^2 = 0.03$). There were two interpretable interactions for DIF recovery with small-medium effect sizes ($0.06 < \eta^2 \leq 0.14$) and they were number of items by LC structure interaction ($\eta^2 = 0.10$) and proportion of DIF by LC structure ($\eta^2 = 0.12$). There was one interpretable interaction for DIF recovery with a large effect size ($\eta^2 > 0.14$) and it was DIF type by LC structure interaction. (There were two interpretable interaction for DIF recovery with medium effect size ($\eta^2 = 0.20$).

Chapter Four: Discussion

It is crucial to ensure that the measure of a construct is invariant between the underlying latent trait and observed scores across subgroups; this is referred to as measurement invariance. Measurement invariance is a premise for an effective and impartial scale. Differential item functioning (DIF) occurs when measurement invariance is violated at the item level, which refers to the mean difference of test scores (non-parametric approaches) or different parameter scores (parametric approaches) among test taker groups after conditioning on the true mean differences among subgroups. Traditional methods of detecting DIF rely on observed covariates such as gender and income level to split subgroups thus ignore the difference within subgroups. The Rasch mixture model (RMM) as an alternative for detecting DIF has the advantage of identifying latent classes (LC) and extracting DIF among LCs.

This study assessed the robustness of the RMM in detecting DIF from two perspectives: latent structure recovery and parameter recovery by fitting RMMs to various simulated datasets of 48 scenarios. The 48 (2*3*2*2*2) scenarios were formed by manipulating five factors: number of items (i.e., test length, 2 levels), proportion of DIF items (2 levels), LC structure (2 levels), group size (2 levels) and DIF type (2 levels).

While prior research suggested use of BIC over AIC in model selection (e.g., Choi et al, 2016; Li et al., 2016), the present study found that when using AIC and BIC to recover the correct latent structure, both information criteria showed a conservative

pattern in which the recovered LCs did not match the true structure perfectly or even in the majority of cases. That is, it was rare that the correct latent structure was recovered at 100%. Specifically, BIC was more conservative than AIC as BIC includes a penalty for a large sample size, which was 3,000 in the current study. Results from the current study contradict those of Li et al (2016) in which they suggested that either BIC or AIC was a reliable statistic in model selection using the Rasch mixture model. However, this study recommends that neither AIC nor BIC should be used as the single decision criterion for determining the true latent structure using a Rasch mixture model. AIC and BIC could be combined with other information criteria such as the Cressie-Read statistic (Read & Cressie, 1988). Unlike AIC and BIC which directly take likelihood ratio into account, the Cressie-Read statistic is based on a constrained minimization of a given function of observations' multinomial probabilities and a nonparametric maximum likelihood estimator, which results in a family of asymptotic test statistics with distributions falling into a family of χ^2 distributions (Bravo, 2002). The Cressie-Read statistic is not available with the current version of the *mirt* package and *pyschomix* on R, but it is available with Winmira software.

Although there were numerous main and interaction effects of the five manipulated factors with small effect sizes ($\eta^2 < 0.06$), their impacts on both LC structure recovery and parameter recovery were limited. As a result, those main and interaction effects with medium to large effect size ($0.06 < \eta^2 < 0.14$) and large effect sizes ($\eta^2 > 0.14$) are the focus of this chapter. In concert with prior research (e.g., DeAyala et al., 2002; Samuelsen, 2005), this study found that group size had an effect ($\eta^2 = 0.11$) on LC

structure recovery, with the RMM reaching a higher correct LC structure recovery rate for equal group size tests than for unequal group size tests. Number of items (i.e., test length) had a large effect on LC structure recovery and longer tests with 30 items had larger AIC and BIC than shorter tests with 10 items. This was predetermined since both AIC and BIC take number of items into account and so 10- and 30- item tests are not nested models.

RMSEs were used as goodness of model fit indices for both classifier recovery and DIF recovery, and lower RMSE indicated better model fit. For classifier recovery, all five manipulated factors showed effect sizes that were medium or larger except DIF type ($\eta^2 < 0.06$). Longer tests with 30 items had better (i.e., lower RMSE) classifier recovery than shorter tests with 10 items. As the proportion of DIF items increased, the performance of the RMM on recovering true classifier increased. Classifiers of two LC structure tests were better recovered via the RMM than those of three LC structure. Tests with equal group size had better classifier recovery than tests with unequal group size, which is consistent with findings of DeAyala et al. (2002) and Preinerstorfer and Formann (2012). There were two medium effect size interactions for classifier recovery, and they were number of items by group size interaction ($\eta^2 = 0.10$) and LC structure by group size interaction ($\eta^2 = 0.13$). When controlling the test length, the performance of the RMM on classifier recovery was better for equal group size tests than unequal group size tests; as number of items increased from 10 items to 30 items, the performance of the RMM on classifier recovery for unequal group size tests improved while it remained constant for equal group size tests. When controlling the LC structure, the performance of

the RMM on classifier recovery was better for equal group size tests than unequal group size tests; as LC structure increased from two LC structure to three LC structure, the performance of the RMM on classifier recovery for unequal group size tests improved while it decreased for equal group size tests. Group size was the only large effect (η^2 = 0.45) for classifier recovery though the interpretation of the effects of group size was complicated by interactions with number of items (η^2 = 0.10) and LC structure (η^2 = 0.13). As the only large main effect and the only factor involving in both medium interactions, group size plays a significant role in classifier recovery. In other words, whether detected latent classes have equivalent group size or not casts an important impact upon classifier recovery. Equal group sizes for detected LCs favor the overall performance of the RMM on classifier recovery. Practitioners can think about adding more items (i.e., test length) to a test to improve the performance of classifier recovery using the RMM to detect DIF. Although, equal size of LCs is desired for better overall classifier recovery, the opposite direction of group size by LC structure interaction direction suggests that unequal group size of LCs to some extent is likely to help improve classifier recovery when there are more than two LCs. The RMM is recommended if the research objective is to explore the number of latent classes from the perspective of classifier recovery, even if the data analyst knows in advance there are unequal group sizes.

There were three main and three interaction medium to large effects of the five manipulated factors on DIF recovery ($\eta^2 > 0.06$) and they were effects of number of items, proportion of DIF items, LC structure, number of items by LC structure

interaction, proportion of DIF items by LC structure interaction, and DIF type by LC structure interaction. Compare to that in classifier recovery, group size had no interpretable effect size on DIF recovery, which is consistent with findings from study conducted by Choi et al (2016) and Li et al (2016). The RMM had better performance on recovering DIF for shorter tests with 10 items than tests with 30 items. As the proportion of DIF items increased, the performance of the RMM on DIF recovery decreased, an opposite direction as with classifier recovery. In concert with classifier recovery, DIF arrays of tests with two LC structure were better recovered than those with three LC structure. As number of items increased from 10 items to 30 items, performance of the RMM on DIF recovery of two LC structure tests remained similar, but that on DIF recovery of three LC structure decreased. As proportion of DIF items increased, performance of the RMM on DIF recovery of two LC structure remained similar, but that on DIF recovery of three LC structure decreased at a constant rate. For symmetric DIF tests, the performance of the RMM on DIF recovery was better for the two LC structure, but for the gradient DIF test, the performance of the RMM on DIF recovery was better for the three LC structure. For every scenario with true DIF of symmetric pattern, means of recovered DIF array of each scenario well restored the symmetric pattern. However, as the proportion of DIF items reached 60%, for tests with gradient DIF, there was a phenomenon of inflation (i.e., the recovered DIFs were consistently larger than true DIFs) of recovered DIF for items with true DIF < 0.3. LC structure had the strongest effect ($\eta^2 = 0.86$) on DIF recovery. Additionally, LC structure was involved in multiple

interactions--with number of items ($\eta^2 = 0.10$), proportion of DIF ($\eta^2 = 0.12$), and DIF type ($\eta^2 = 0.20$).

So, while group size ($\eta^2 = 0.45$) had the strongest effect on classifier recovery with LC structure ($\eta^2 = 0.08$) having a medium effect, for DIF recovery LC structure ($\eta^2 = 0.86$) had by far the strongest effect while group size ($\eta^2 = 0.001$) had no interpretable effect. Group size and LC structure (i.e., number of latent classes) define the characteristics of latent classes from observed response patterns of a measurement construct. This implies that accuracy of detected proportion of latent classes is strongly influenced by true latent class size and its interaction with number of latent classes, but the accuracy of DIF detection using the RMM is influenced by number of latent classes and has nearly no relation with group size of each latent class.

Taking everything into account, some conclusions are provided from this study for practitioners when using the Rasch mixture model in detecting differential item functioning. The RMM is more likely to obtain the best recovery of classifier parameters or, in other words, identify the closest proportion array to the true proportion array, when LCs are close to each other in group size. However, group size has little influence on performance of the RMM on detecting true DIF among LCs. Instead, number of items of a test, proportion of DIF items, and LC structure of observations play important roles on DIF recovery using the RMSE. Specifically, LC structure had by far the strongest effect on DIF recovery. The optimal scenario for obtaining recovered DIF closer to true DIF is for, tests with number of items close to 10, proportion of DIF close to 20%, and likely to

have a two LC structure. DIF type was found to have only a small main effect size for both classifier recovery ($\eta^2 = 0.06$) and DIF recovery ($\eta^2 = 0.05$).

However, DIF type by LC structure in DIF recovery was the only large interaction effect ($\eta^2 = 0.20$) among all the interpretable interaction effects. The principal difference between a gradient DIF pattern and a symmetric is the former has zero DTF while the latter has non-zero DTF. Gradient DIF pattern improved the performance of the RMM on DIF recovery of three LC structure tests compared to that of two LC structure. In contrast, symmetric DIF pattern improves the performance of the RMM on DIF recovery of two LC structure tests compare to that of three LC structure.

When using an RMM to determine DIF, it is recommended to have close group sizes for latent classes, 20% to 40% proportion of DIF items and a LC structure close to a two LC structure. Since the AIC and BIC are not suggested in this study, the Cressie-Read statistic, as an alternative, can be used as a model selection tool for picking the model with correct number of latent classes. This study suggests trusting detected group size if the ratio of group size between two latent classes is larger than 0.5 but smaller than 2, which is considered as a close group size. The interpretation of LCs is critical for using the RMM to detect DIF, especially when there are multiple LCs. It is recommended to associate detected LCs and covariates of interest and examine the overlap and intersectionality between LCs and covariates. The more LCs a covariate (e.g., gender) involves, the more intra-group differentiation there is based on the observed response pattern. A practitioner can identify DIF and its direction through calculating the item difficulty difference Δb between two latent classes. It can be considered as no item DIF

for using the RMM method when $\Delta b < 0.3$, small DIF when $0.3 \leq \Delta b < 0.9$, medium DIF when $0.9 \leq \Delta b < 1.5$, and large DIF when $\Delta b \geq 1.5$.

Based on the functionality and mechanism of the RMM, a suggested analytic path for using the RMM to detect LCs and DIF among LCs will be (1) check the unidimensionality assumption of the RMM, (2) fit several RMMs with a different number of possible LCs (e.g., k = 1, 2, 3), (3) pick the best fitting model based on certain model selection information criteria (e.g., Cressie-Read), (4) calculate DIF among detected LCs and summarize the magnitude of DIF according to above DIF cutoffs, (5) interpret the meaning of LCs combining the overlap between the LC and covariates (e.g., gender, education level, training) and (6) interpret any DIF that is relevant to answer the research question. If the result of using the RMM to detect LCs and DIF was unsatisfactory or unclear, practitioners are encouraged to refer to another LC detection tools such as 2PL mixture model or Gaussian mixture model.

Limitations

Although it performed well on two LC structure tests, the Rasch mixture model's performance on DIF recovery of three LC structure tests was far from ideal. This study used fixed latent ability (θ) differences (i.e., impact = 1) among LCs to afford quick convergence of the model, as was supported by the previous literature. For a two LC structure, latent traits of reference LC and focal LC were from N (0, 1) and N (1, 1) respectively, and for a three LC structure, the latent trait of reference LC and focal LCs were from N(0, 1), N (1, 1), and N (-1, 1). However, it is unclear whether there is a difference between zero-impact settings, which mean the latent trait of LCs come from

the same normal distribution with mean of 0 and SD of 1, and current settings on latent structure recovery and parameter recovery via the Rasch mixture model.

Additionally, because this study only included a two LC structure and a three LC structure, the power of generalization of the RMM on detecting DIF is limited to those two conditions. There are multiple other factors worth investigating that have not been included in this study such as item type (e.g., polynomial item). The computation time for replications in this study was relatively long, about four days in this study, thus it is likely to be interrupted by random factors such as power outage and overheating of the CPU. Because computation time was lengthy, especially for LC structure recovery, this study used 100 replications for each scenario of LC structure recovery but used 200 replications for each scenario of DIF recovery. Both AIC and BIC are likely to be inadequate indices for structure recovery using the RMM on DIF detection.

Tests with 10 items or 30 items which were examined in the current study are relatively short from an educational assessment perspective. Results from the current study may not apply to typical educational assessments with much longer test lengths. Furthermore, as the test length increases, fatigue of test taker is likely to grow which can cause random response and affect validity of corresponding construct.

Future Research

Future research could include a one LC structure as a baseline for examining the effectiveness of the Rasch mixture model on detecting LC structure and recovering DIF. Similar to positing a unidimensional model underlying an item set, comparison to the simplest structure of a single latent class seems warranted and is more in concert with the

Rasch idea of a “ruler” for measurement that applies universally. Additionally, this complies with a rule of parsimony as including more latent classes result in estimating more parameters for fitting a Rasch mixture model. The findings of this study could also provide direction for a psychometrician who is interested in creating a new measure. She or he should be aware of within covariate-based subgroup difference which denote the existence of latent classes and cautiously design the covariate data being collected based on targeted participants and research objectives. That is, psychometricians may posit the existence of multiple latent classes in some circumstances and deliberately include targeted covariates in the data collection plan to ensure the ability to interpret the latent classes if they are found.

Finding more effective information indices for the Rasch mixture model on DIF detection is a topic for future study. Information criteria that directly use likelihood ratios such as BIC and AIC proved to be ineffective in extracting the correct LC structure in this study. Both AIC and BIC are likely to underestimate the true number of latent classes. Future study could modify the current AIC or BIC formulas to accommodate the usage of the Rach mixture model to detecting the latent class structure from a set of responses. Additionally, the Cressie-Read statistic seems a possible alternative as it incorporates a statistical test through transforming model fit indices. Future study could compare the Cressie-Read statistic and information criteria which directly use a likelihood ratio on the accuracy of detecting latent classes using the Rasch mixture model.

Future study can look into tests with much longer test length than 30 items because those tests are very common for educational assessment. Fatigue of test takers is a crucial issue, especially for tests with long lengths, and it is a problem which is difficult to quantify. So, a mixed method approach combining both qualitative and quantitative methods can be an option for researching the robustness of the Rasch mixture model to detect DIF on typical educational assessments having more than 30 items. Latent classes formed with measures of perseverance or distractibility may be useful in helping to determine whether fatigue is a crucial problem.

Advanced computation tools such as parallel computing could be used to increase the efficiency of conducting simulation via the EM algorithm. The current data generation and simulation codes using R are very time-consuming and thus rely on the availability of computation power. Future study could explore the potential algorithm of the Rasch mixture model simulation with a more efficient programming language such as C++.

The number of replications used for LC structure recovery and parameter recovery used in this study (100 and 200) are recommended for future study on a similar topic, because the model fitting reached convergence with relative stable RMSE. Compared to other IRT based DIF detection methods such as LRT and Lord's χ^2, the Rasch mixture model would be suggested to detect LC structure and find DIF among latent classes when there are likely to be only two or three latent classes.

Nearly all IRT based DIF detection methods including the Rasch mixture model use parameter differences among subgroups as the effect size of DIF. However, whether

this is a reliable measure of DIF was not addressed by the current study and is a topic for future study. Further, future research may address other ways of calculating DIF and compare those results to the current approach.

References

Adams, R. J., Wu, M. L., & Carstensen, C. H. (2007). Application of multivariate Rasch models in international large-scale educational assessments. In M. von Davier, & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models: extensions and applications* (pp. 271-280). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_17 Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-017&site=ehost-live&scope=site

Alexeev, N., Templin, J., & Cohen, A. S. (2011). Spurious latent classes in the mixture Rasch model. *Journal of Educational Measurement, 48*(3), 313-332. doi:http://dx.doi.org.du.idm.oclc.org/10.1111/j.1745-3984.2011.00146.x

Aryadoust, V. (2015). Fitting a mixture Rasch model to English as a foreign language listening tests: The role of cognitive and background variables in explaining latent differential item functioning. *International Journal of Testing, 15*(3), 216-238. doi:10.1080/15305058.2015.1004409

Bolt, D. M., Cohen, A. S., & Wollack, J. A. (2002). Item parameter estimation under conditions of test speediness: Application of a mixture Rasch model with ordinal constraints. *Journal of Educational Measurement, 39*(4), 331-348. doi:10.1111/j.1745-3984.2002.tb01146.x

Bravo, F. (2002). Blockwise empirical Cressie–Read test statistics for α-mixing processes. Statistics & Probability Letters, 58(3), 319-325. doi:10.1016/S0167-7152(02)00162-1

Byrne, B. M., Baron, P., & Campbell, T. L. (1994). The Beck Depression Inventory (French version): testing for gender-invariant factorial structure for nonclinical adolescents. *Journal of Adolescent Research*, *9*(2), 166–179. doi: https://doi.org/10.1177/074355489492003

Cohen, A. S., & Kim, S. (2011). Detecting cognitive change in the math skills of low-achieving adolescents. *The Journal of Special Education, 45*(2), 67-76. doi:10.1177/0022466909351579

Cho, Y. (2013). *The mixture distribution polytomous Rasch model used to account for response styles on rating scales: A simulation study of parameter recovery and classification accuracy* (Ph.D.). Available from ProQuest Central, ProQuest Dissertations & Theses Global, Social Science Premium Collection. (1462059411). Retrieved from https://du.idm.oclc.org/login?url=https://search-proquest-com.du.idm.oclc.org/docview/1462059411?accountid=14608

Choi, I., Paek, I., & Cho, S. (2017). The impact of various class-distinction features on model selection in the mixture Rasch model. *Journal of Experimental Education, 85*(3), 411-424. doi:http://dx.doi.org.du.idm.oclc.org/10.1080/00220973.2016.1250208

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, N.J.: L. Erlbaum Associates.

Dai, Y. (2013). A mixture Rasch model with a covariate: A simulation study via Bayesian Markov chain Monte Carlo estimation. *Applied Psychological Measurement, 37*(5), 375-396. doi:10.1177/0146621612475076

De Ayala, R., J., Kim, S., Stapleton, L. M., & Dayton, C. M. (2002). Differential item functioning: A mixture distribution conceptualization. International Journal of Testing, 2(3-4), 243-276. doi:10.1080/15305058.2002.9669495

Demars, C. E., & Lau, A. (2011). Differential item functioning detection with latent classes: How accurately can we detect who is responding differentially? *Educational and Psychological Measurement, 71*(4), 597-616. doi:10.1177/0013164411404221

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1-22. doi:10.1111/j.2517-6161.1977.tb01600.x

Eid, M., & Zickar, M. J. (2007). Detecting response styles and faking in personality and organizational assessments by mixed Rasch models. In M. von Davier, & C. H. Carstensen (Eds.), (pp. 255-270). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_16 Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-016&site=ehost-live&scope=site

Frederickx, S., Tuerlinckx, F., De Boeck, P., & Magis, D. (2010). RIM: A random item mixture model to detect differential item functioning. *Journal of Educational Measurement, 47*(4), 432-457. doi:http://dx.doi.org.du.idm.oclc.org/10.1111/j.1745-3984.2010.00122.x

Frick, H., Strobl, C., & Zeileis, A. (2015). Rasch mixture models for DIF detection: A comparison of old and new score specifications. *Educational and Psychological*

Measurement, 75(2), 208-234. doi: http://dx.doi.org.du.idm.oclc.org/10.1177/0013164414536183

Glück, J., & Spiel, C. (2007). Studying development via item response models: A wide range of potential uses. In M. von Davier, & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models: extensions and applications* (pp. 281-292). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_18 Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-018&site=ehost-live&scope=site

Izsák, A., Orrill, C. H., Cohen, A. S., & Brown, E. R. (2010). Measuring middle grades teachers' understanding of rational numbers with the mixture Rasch model. *The Elementary School Journal, 110*(3), 279-300. doi:10.1086/648979

Kroenke, K., Spitzer, R. L., & Williams, J. B. W. (2001). The PHQ-9: Validity of a brief depression severity measure. *Journal of General Internal Medicine*; *16*(9), 606-613. doi:10.1046/j.1525-1497.2001.016009606.x

Mortimer, R. (2008). Marketing theory: treasure seekers. *Brand Strategy*, , 25-29. Retrieved from https://du.idm.oclc.org/login?url=https://www-proquest-com.du.idm.oclc.org/docview/224194839?accountid=14608

Jang, Y., Kim, S., & Cohen, A. S. (2018). The impact of multidimensionality on extraction of latent classes in mixture Rasch models. *Journal of Educational Measurement, 55*(3), 403-420. doi:http://dx.doi.org.du.idm.oclc.org/10.1111/jedm.12185

Jiao, H., Lissitz, R. W., Macready, G., Wang, S., & Liang, S. (2011). Exploring levels of performance using the mixture Rasch model for standard setting. *Psychological Test and Assessment Modeling, 53*(4), 499-522. Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2012-03365-006&site=ehost-live&scope=site

Lee, H., & Beretvas, S. N. (2014). Evaluation of two types of differential item functioning in factor mixture models with binary outcomes. *Educational and Psychological Measurement, 74*(5), 831-858. doi:10.1177/0013164414526881

Li, F., Cohen, A. S., Kim, S., & Cho, S. (2009). Model selection methods for mixture dichotomous IRT models. *Applied Psychological Measurement, 33*(5), 353-373. doi:10.1177/0146621608326422

Li, T., Jiao, H., & Macready, G. B. (2016). Different approaches to covariate inclusion in the mixture Rasch model. *Educational and Psychological Measurement, 76*(5), 848-872. doi:10.1177/0013164415610380

Li, Y., Jiao, H., & Lissitz, R. (2014). Applying multidimensional item response theory models in validating test dimensionality: An example of K–12 large-scale science assessment. *Journal of Applied Testing Technology, 13*(2). Retrieved from http://www.jattjournal.com/index.php/atp/article/view/48367

Maij-de Meij, A., M., Kelderman, H., & van Der Flier, H. (2008). Fitting a mixture item response theory model to personality questionnaire data: Characterizing latent classes and investigating possibilities for improving prediction. *Applied Psychological Measurement, 32*(8), 611-631. doi:10.1177/0146621607312613

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Read T.R.C., & Cressie N.A.C. (1988) Introduction to the Power-Divergence Statistic. In: Goodness-of-Fit Statistics for Discrete Multivariate Data. Springer Series in Statistics. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-4578-0_1

Rijkes, C. P. M., & Kelderman, H. (2007). Latent-response Rasch models for strategy shifts in problem-solving processes. In M. von Davier, & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models: extensions and applications* (pp. 311-328). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_20 Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-020&site=ehost-live&scope=site

Rost, J. (1990). Rasch models in latent classes: An integration of two approaches to item analysis. *Applied Psychological Measurement, 14*(3), 271–282. https://doi.org/10.1177/014662169001400305

Tenenbaum, G., Strauss, B., & BÃ¼sch, D. (2007). Applications of generalized Rasch models in the sport, exercise, and the motor domains. In M. von Davier, & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models: extensions and applications* (pp. 347-356). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_22 Retrieved

from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-022&site=ehost-live&scope=site

Toker, T. (2017). *A comparison of latent class analysis and the mixture Rasch model: A cross-cultural comparison of 8th grade mathematics achievement in the fourth international mathematics and science study (TIMSS-2011)* (Publication No. 10163571) [Doctoral dissertation, University of Denver]. ProQuest Dissertations & Theses Global; SciTech Premium Collection; Social Science Premium Collection. (1830757654). Retrieved from https://du.idm.oclc.org/login?url=https://www-proquest-com.du.idm.oclc.org/docview/1830757654?accountid=14608

von Davier, M., & Yamamoto, K. (2007). Mixture-distribution and HYBRID Rasch models. In M. von Davier, & C. H. Carstensen (Eds.), *Multivariate and mixture distribution Rasch models: extensions and applications* (pp. 99-115). NY: Springer Science + Business Media. doi:10.1007/978-0-387-49839-3_6 Retrieved from http://du.idm.oclc.org/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2007-06902-006&site=ehost-live&scope=site

Weiss, D. J. (1982). Improving measurement quality and efficiency with adaptive testing. *Applied Psychological Measurement, 6*(4), 473-492. doi:10.1177/014662168200600408

Wu, P., & Huang, T. (2010). Person heterogeneity of the BDI-II-C and its effects on dimensionality and construct validity: Using mixture item response models. *Measurement and Evaluation in Counseling and Development, 43*(3), 155-167. doi:http://dx.doi.org.du.idm.oclc.org/10.1177/0748175610384808

Wu, Y., & Paek, I. (2018). Agreement on the classification of latent class classifier between different identification constraint approaches in the mixture Rasch model. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences, 14*(2), 82-93. doi:10.1027/1614-2241/a000148

Wyse, A. E. (2019). Analyzing job analysis data using mixture Rasch models. *International Journal of Testing, 19*(1), 52–73. https://doi-org.du.idm.oclc.org/10.1080/15305058.2018.1481853

Appendices

Appendix A Codes for Latent Class Structure Recovery and Parameter Recovery

```{r, warning = FALSE}
library(mirt)
library(psychomix)
```

Generating Datasets

Generating Parameters

```{r}
# List delta_b for symmetric DIF patterns
d_b_1002s <- c(-1.8, 1.8, 0, 0, 0, 0, 0, 0, 0 ,0)
d_b_1004s <- c(-1.8, -0.9, 0.9, 1.8, 0, 0 ,0, 0, 0, 0)
d_b_1006s <- c(-1.8, -1.2, -0.6, 0.6, 1.2 ,1.8, 0, 0 ,0 ,0)
d_b_3006s <- c(-1.8, -1.2, -0.6, 0.6, 1.2 ,1.8, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0)
d_b_3012s <- c(-1.8, -1.5, -1.2, -0.9, -0.6, -0.3, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8,0, 0, 0 ,0 ,0, 0, 0, 0 ,0 ,0 ,0, 0, 0 ,0 ,0 ,0, 0, 0, 0)
d_b_3018s <- c(-1.8, -1.6, -1.4, -1.2, -1.0, -0.8, -0.6, -0.4, -0.2, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 0, 0 ,0 ,0, 0, 0 ,0 ,0, 0, 0 ,0 ,0)

# list delta_b for gradient DIF patterns
d_b_1002g <- c(2.0, 1.0, 0, 0, 0, 0, 0, 0, 0 ,0)
d_b_1004g <- c(2.0, 1.5, 1.0, 0.5, 0, 0, 0, 0, 0 ,0)
d_b_1006g <- c(2.0, 1.7, 1.4, 1.1, 0.8, 0.5, 0, 0, 0, 0)
d_b_3006g <- c(2.0, 1.7, 1.4, 1.1, 0.8, 0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
d_b_3012g <- c(2.0, 1.8, 1.6, 1.4, 1.2, 1.0, 0.8, 0.6, 0.4, 0.3, 0.2, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
d_b_3018g <- c(1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

# Create 4 variables to store DIF patterns
s_dif_10items <- cbind(d_b_1002s, d_b_1004s, d_b_1006s)
s_dif_30items <- cbind(d_b_3006s, d_b_3012s, d_b_3018s)

g_dif_10items <- cbind(d_b_1002g, d_b_1004g, d_b_1006g)
g_dif_30items <- cbind(d_b_3006g, d_b_3012g, d_b_3018g)
```

Responses Generating Function
```{r}
gen_response <- function(n_item, size, dif){
  b_0 <- matrix(rnorm(n_item, 0, 1))
  sigma <- matrix(1)
  type <- rep('dich', n_item)
  a <- matrix(rep(1, n_item))
  df = ''
  n_lc = length(size)
  if (n_lc == 2){
    df_lcr = simdata(a = a, d = b_0, itemtype = type, N = size[1], mu = 0, sigma = sigma)
    b_lc1 = b_0 + dif
    df_lcf1 = simdata(a = a, d = b_lc1, itemtype = type, N = size[2], mu = 1, sigma = sigma)
    #combine reference lc and focal lc datasets
    df = as.data.frame(rbind(df_lcr, df_lcf1))
    #add identifiers for each row/observation
    df$lc = c(rep('lc_r', size[1]), rep('lc_f1', size[2]))
  } else{
    df_lcr = simdata(a = a, d = b_0, itemtype = type, N = size[1], mu = 0, sigma = sigma)
    b_lc1 = b_0 + dif
    b_lc2 = b_0 + 2*dif
    df_lcf1 = simdata(a = a, d = b_lc1, itemtype = type, N = size[2], mu = 1, sigma = sigma)
    df_lcf2 = simdata(a = a, d = b_lc2, itemtype = type, N = size[3], mu = -1, sigma = sigma)
    #combine lc_r and lc_f1 and lc_f2
    df = as.data.frame(rbind(df_lcr, df_lcf1, df_lcf2))
    #add identifiers
    df$lc = c(rep('lc_r', size[1]), rep('lc_f1', size[2]), rep('lc_f2', size[3]))
  }
return(df)
}

```


Parameter Recovery Simulation Function
```{r}
par_reco_sim <- function(dif, size_of_lc){

  set.seed(123)

  # Set number of replications
  n_rep = 200
```

```
 # Set number of items
 n_item = length(dif)
 # Number of latent classes
 n_lc = length(size_of_lc)
 # Create a matrix to store item difficulty parameters for every replication
 dif_matrix = matrix(NA, n_rep, n_item)
 # Matrix for 2*dif for third latent class
 dif_matrix_2 = matrix(NA, n_rep, n_item)
 # Create a matrix to store proportion of latent classes
 p_matrix = matrix(NA, n_rep, n_lc)

##########################################################################
#####

 if (n_lc == 2){
  # Simulation Loop
  for (rep in 1: n_rep){
   test = gen_response(n_item, size_of_lc, dif)
   m <- raschmix(as.matrix(test[, 1:n_item]), k = n_lc, scores = "saturated", nrep = 1)
   results = worth(m)
   for(j in 1:n_item){
    dif_matrix[rep, j] = abs(results[j, 2] - results[j, 1])   # Get DIF after a replication
   }
   p_matrix[rep, 1] = parameters(m, which = 'concomitant')[1] # Record proportion of
latent classes
   p_matrix[rep, 2] = parameters(m, which = 'concomitant')[2] # Record proportion of
latent classes
  }
  # Stop the clock and sotre runtime in rt
  rt = (proc.time() - ptm)[1:3]
  outcome = cbind(dif_matrix, p_matrix)
  return(outcome)
 }
 else{
  for (rep in 1: n_rep){
   test = gen_response(n_item, size_of_lc, dif)
   m <- raschmix(as.matrix(test[, 1:n_item]), k = n_lc, scores = "saturated", nrep = 1)
   results = worth(m)
   for(j in 1:n_item){
    dif_matrix[rep, j] = abs(results[j, 2] - results[j, 1])   # Get DIF after a replication
    dif_matrix_2[rep, j] = abs(results[j, 3] - results[j, 1]) # Get 2*DIF between third
latent class and referenc latent class
```

```
    }
    p_matrix[rep, 1] = parameters(m, which = 'concomitant')[1] # Record proportion of
latent classes
    p_matrix[rep, 2] = parameters(m, which = 'concomitant')[2] # Record proportion of
latent classes
    p_matrix[rep, 3] = parameters(m, which = 'concomitant')[3] # Record proportion of
latent classes
   }
   outcome = cbind(dif_matrix, dif_matrix_2, p_matrix)
   return(outcome)
 }
}
```

DIF Visualization Function

```{r}
dif_viz <- function (dif, dif_sim_matrix){
 # Number of DIF item
 n_dif = length(which(dif != 0)) # only graph DIF items

 if (n_dif %% 6 == 0){
  if (n_dif==18){
   par(mfrow=c(3, 3))
  } else {
  par(mfrow=c(2, 3))
  }
 } else {
  if (n_dif==2){
   par(mfrow=c(1, 2))
  } else {
   par(mfrow=c(2, 2))
  }
 }
 for(i in 1: n_dif){
  true_dif = abs(dif[i])

  plot(c(1:nrow(dif_sim_matrix)), dif_sim_matrix[, i], type = 'l', ylim = c(0,5), xlab =
'Replication', ylab = 'Recovered DIF', main = paste('Item', toString(i),'True
DIF:',toString(true_dif)), cex.lab=1.2, cex.main=1.2, font.main=7)

  # Add a red horizotal line marking mean of DIF from replications
```

```
   abline(h = mean(dif_sim_matrix[,i]), col = 'red')

   # Add a green horizontal line marking true DIF
   abline(h = true_dif, col = 'green')

   # Add lengend
   legend('top', legend = c('Mean DIF','True DIF'), lty = 1, col = c('red', 'green'), lwd=2,
cex=1, y.intersp=0.5, x.intersp = 0.12, horiz = TRUE, bty="n")
  }
}
```

Classifier Parameter Visualization Function

```{r}
p_viz <- function (p_matrix, size){
  if (length(size) == 2){
   # True proportion
   true_p <- round(size[1]/sum(size), 3)
   p_prior <- apply(p_matrix, 1, max)
   plot(x = c(1:nrow(p_matrix)),y = p_prior, type = "l", ylim = c(0, 1), xlab = "Replication",
ylab = 'Proporiton of Latent Class', main = paste('Two Latent Classes, True Proportion:',
toString(true_p)))
   abline(h = true_p, col = 'green')
   abline(h = mean(p_prior), col = 'red')
   legend('topright', legend = c('Mean Proportion','True Proportion'), lty = 1, col = c('red',
'green'))
  }
  else{
   true_p <- c(round(size[1]/sum(size), 2), round(size[3]/sum(size), 2))
   p_prior <- cbind( max_p = apply(p_matrix, 1, max), min_p = apply(p_matrix, 1, min))
   plot(x = c(1:nrow(p_matrix)),y = p_prior[,1], type = "l", ylim = c(0, 1), xlab =
"Replication", ylab = 'Proporiton of Latent Class', main = paste('Three Latent Classes,
True Proportion:', toString(true_p[1]), toString(true_p[2])))
   points(x = c(1:nrow(p_matrix)),y = p_prior[,2], type = "l")
   abline(h = true_p[1], col = 'green', lty = 1)
   abline(h = true_p[2], col = 'green', lty = 2)
   abline(h = mean(p_prior[,1]), col = 'red', lty = 1)
   abline(h = mean(p_prior[,2]), col = 'red', lty = 2)
   legend('top', legend = c('Ture Max Proportion','True Min Prorpotion', 'Mean Max
Proportion', 'Mean Min Proportion'), col = c('green', 'green', 'red', 'red'), lty = c(1, 2, 1,
2))
```

```
 }
}
```

```
##############################################################################
######
###############            Latent Class Structure Recovery
############################
##############################################################################
######
```

```
############### Set number of replications for LC structure recovery below
################
```
```{r}
n_rep = 100
```

```
## Function for calculating latent structure recovery rate from generated results
dataframe
```
```{r}
get_structure_reco_rate <- function (df, lc_Structure) {

 ## update index to have logical order
 row.names(df) = seq(1, nrow(df))

 n = length(lc_Structure)
 count_AIC = 0
 count_BIC = 0

 aic_and_bic <- matrix(rep(0, 4), nrow=2, ncol=4)
 colnames(aic_and_bic) <- c(1:4)
 rownames(aic_and_bic) <- c("AIC", "BIC")

 for (i in seq(1, nrow(df), 4)) {
  start = i
  end = i + 3
  temp_df = df[start:end,]
```

```
   k_temp = temp_df$k[which.min(temp_df$AIC)]
   as.character(k_temp)
   aic_and_bic["AIC", k_temp] = aic_and_bic["AIC", k_temp] + 1

   k_temp = temp_df$k[which.min(temp_df$BIC)]
   as.character(k_temp)
   aic_and_bic["BIC", k_temp] = aic_and_bic["BIC", k_temp] + 1

 }

 aic_and_bic = aic_and_bic/n_rep

 return (aic_and_bic)
}
```

########## TWO LATENT CLASSES STRUCTURE
##

######## Model Recovery - Symmetric Pattern - Two Latent Classes - Equal Size
############

Model Recovery 10 Items 2 DIF Symmetric Pattern: 1002s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1002s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1002s_e = list()
```

```
for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1002s_e = rbind(mf_lc2_1002s_e, show(m2))
}

## calculate recovery rate
lcReco_1002s_lc2_e = get_structure_reco_rate(mf_lc2_1002s_e, size_of_lc)

```


Model Recovery 10 Items 4 DIF Symmetric Pattern: 1004s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1004s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1004s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1004s_e = rbind(mf_lc2_1004s_e, show(m2))
}

## calculate recovery rate
lcReco_1004s_lc2_e = get_structure_reco_rate(mf_lc2_1004s_e, size_of_lc)

```

Model Recovery 10 Items 6 DIF Symmetric Pattern: 1006s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1006s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1006s_e = rbind(mf_lc2_1006s_e, show(m2))
}

## calculate recovery rate
lcReco_1006s_lc2_e = get_structure_reco_rate(mf_lc2_1006s_e, size_of_lc)

```

Model Recovery 30 Items 6 DIF Symmetric Pattern: 3006s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3006s
```

```
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3006s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3006s_e = rbind(mf_lc2_3006s_e, show(m2))
}

## calculate recovery rate
lcReco_3006s_lc2_e = get_structure_reco_rate(mf_lc2_3006s_e, size_of_lc)
```

Model Recovery 30 Items 12 DIF Symmetric Pattern: 3012s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3012s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3012s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3012s_e = rbind(mf_lc2_3012s_e, show(m2))
}

## calculate recovery rate
```

```
lcReco_3012s_lc2_e = get_structure_reco_rate(mf_lc2_3012s_e, size_of_lc)

```

Model Recovery 30 Items 18 DIF Symmetric Pattern: 3018s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 30
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3018s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3018s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3018s_e = rbind(mf_lc2_3018s_e, show(m2))
}

## calculate recovery rate
lcReco_3018s_lc2_e = get_structure_reco_rate(mf_lc2_3018s_e, size_of_lc)

```

Model Recovery - Symmetric Pattern - Two Latent Classes - Unequal Size
###############

Model Recovery 10 Items 2 DIF Symmetric Pattern: 1002s - unequal size
```{r}
set.seed(12345)

# Set number of replications
```

```
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_1002s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1002s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1002s_u = rbind(mf_lc2_1002s_u, show(m2))
}

## calculate recovery rate
lcReco_1002s_lc2_u = get_structure_reco_rate(mf_lc2_1002s_u, size_of_lc)
```

```
## Model Recovery 10 Items 4 DIF Symmetric Pattern: 1004s - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_1004s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1004s_u = list()
```
```

```
for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1004s_u = rbind(mf_lc2_1004s_u, show(m2))
}

## calculate recovery rate
lcReco_1004s_lc2_u = get_structure_reco_rate(mf_lc2_1004s_u, size_of_lc)

```

Model Recovery 10 Items 6 DIF Symmetric Pattern: 1006s - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_1006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1006s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1006s_u = rbind(mf_lc2_1006s_u, show(m2))
}

## calculate recovery rate
lcReco_1006s_lc2_u = get_structure_reco_rate(mf_lc2_1006s_u, size_of_lc)

```

Model Recovery 30 Items 6 DIF Symmetric Pattern: 3006s - unequal size
```{r}
```

```
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3006s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3006s_u = rbind(mf_lc2_3006s_u, show(m2))
}

## calculate recovery rate
lcReco_3006s_lc2_u = get_structure_reco_rate(mf_lc2_3006s_u, size_of_lc)
```

Model Recovery 30 Items 12 DIF Symmetric Pattern: 3012s - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 3
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3012s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
```

```
mf_lc2_3012s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3012s_u = rbind(mf_lc2_3012s_u, show(m2))
}

## calculate recovery rate
lcReco_3012s_lc2_u = get_structure_reco_rate(mf_lc2_3012s_u, size_of_lc)

```

Model Recovery 30 Items 18 DIF Symmetric Pattern: 3018s - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 3
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3018s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3018s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3018s_u = rbind(mf_lc2_3018s_u, show(m2))
}

## calculate recovery rate
lcReco_3018s_lc2_u = get_structure_reco_rate(mf_lc2_3018s_u, size_of_lc)

```

########### Model Recovery - Gradient Pattern - Two Latent Classes - Equal Size ###########

Model Recovery 10 Items 2 DIF Gradient Pattern: 1002g - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1002g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1002g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1002g_e = rbind(mf_lc2_1002g_e, show(m2))
}

## calculate recovery rate
lcReco_1002g_lc2_e = get_structure_reco_rate(mf_lc2_1002g_e, size_of_lc)

```

Model Recovery 10 Items 4 DIF Gradient Pattern: 1004g - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
```

```
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1004g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1004g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1004g_e = rbind(mf_lc2_1004g_e, show(m2))
}

## calculate recovery rate
lcReco_1004g_lc2_e = get_structure_reco_rate(mf_lc2_1004g_e, size_of_lc)

```

Model Recovery 10 Items 6 DIF Gradient Pattern: 1006g - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_1006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1006g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
```

```
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1006g_e = rbind(mf_lc2_1006g_e, show(m2))
}

## calculate recovery rate
lcReco_1006g_lc2_e = get_structure_reco_rate(mf_lc2_1006g_e, size_of_lc)

```

Model Recovery 30 Items 6 DIF Gradient Pattern: 3006g - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3006g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3006g_e = rbind(mf_lc2_3006g_e, show(m2))
}

## calculate recovery rate
lcReco_3006g_lc2_e = get_structure_reco_rate(mf_lc2_3006g_e, size_of_lc)

```

Model Recovery 30 Items 12 DIF Gradient Pattern: 3012g - equal size
```{r}
set.seed(12345)
```

```
# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3012g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3012g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3012g_e = rbind(mf_lc2_3012g_e, show(m2))
}

## calculate recovery rate
lcReco_3012g_lc2_e = get_structure_reco_rate(mf_lc2_3012g_e, size_of_lc)

```

Model Recovery 30 Items 18 DIF Gradient Pattern: 3018g - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 30
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1500)
# Set DIF type
dif_array = d_b_3018g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3018g_e = list()
```

```
for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3018g_e = rbind(mf_lc2_3018g_e, show(m2))
}

## calculate recovery rate
lcReco_3018g_lc2_e = get_structure_reco_rate(mf_lc2_3018g_e, size_of_lc)
```

```
######### Model Recovery - Gradient Pattern - Two Latent Classes - Unequal Size
##########


## Model Recovery 10 Items 2 DIF Gradient Pattern: 1002g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_1002g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1002g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1002g_u = rbind(mf_lc2_1002g_u, show(m2))
}

## calculate recovery rate
lcReco_1002g_lc2_u = get_structure_reco_rate(mf_lc2_1002g_u, size_of_lc)
```
```

```

## Model Recovery 10 Items 4 DIF Gradient Pattern: 1004g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_1004g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1004g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1004g_u = rbind(mf_lc2_1004g_u, show(m2))
}

## calculate recovery rate
lcReco_1004g_lc2_u = get_structure_reco_rate(mf_lc2_1004g_u, size_of_lc)

```

Model Recovery 10 Items 6 DIF Gradient Pattern: 1006g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(2000, 1000)
```
```

```
# Set DIF type
dif_array = d_b_1006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_1006g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_1006g_u = rbind(mf_lc2_1006g_u, show(m2))
}

## calculate recovery rate
lcReco_1006g_lc2_u = get_structure_reco_rate(mf_lc2_1006g_u, size_of_lc)
```

Model Recovery 30 Items 6 DIF Gradient Pattern: 3006g - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3006g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3006g_u = rbind(mf_lc2_3006g_u, show(m2))
}
```

```
## calculate recovery rate
lcReco_3006g_lc2_u = get_structure_reco_rate(mf_lc2_3006g_u, size_of_lc)

```

Model Recovery 30 Items 12 DIF Gradient Pattern: 3012g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3012g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3012g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3012g_u = rbind(mf_lc2_3012g_u, show(m2))
}

## calculate recovery rate
lcReco_3012g_lc2_u = get_structure_reco_rate(mf_lc2_3012g_u, size_of_lc)

```

Model Recovery 30 Items 18 DIF Gradient Pattern: 3018g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
```

```
n_item = 30
# Set sample size
size_of_lc = c(2000, 1000)
# Set DIF type
dif_array = d_b_3018g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc2_3018g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc2_3018g_u = rbind(mf_lc2_3018g_u, show(m2))
}

## calculate recovery rate
lcReco_3018g_lc2_u = get_structure_reco_rate(mf_lc2_3018g_u, size_of_lc)

```


############## THREE LATENT CLASSES STRUCTURE
#####################################

Model Recovery - Symmetric Pattern - Three Latent Classes - Equal Size
##############

Model Recovery 10 Items 2 DIF Symmetric Pattern: 1002s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1002s
# Set number of latent classes
```

```
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1002s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1002s_e = rbind(mf_lc3_1002s_e, show(m2))
}

## calculate recovery rate
lcReco_1002s_lc3_e = get_structure_reco_rate(mf_lc3_1002s_e, size_of_lc)
```

Model Recovery 10 Items 4 DIF Symmetric Pattern: 1004s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1004s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1004s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1004s_e = rbind(mf_lc3_1004s_e, show(m2))
}

## calculate recovery rate
lcReco_1004s_lc3_e = get_structure_reco_rate(mf_lc3_1004s_e, size_of_lc)
```

```

## Model Recovery 10 Items 6 DIF Symmetric Pattern: 1006s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1006s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1006s_e = rbind(mf_lc3_1006s_e, show(m2))
}

## calculate recovery rate
lcReco_1006s_lc3_e = get_structure_reco_rate(mf_lc3_1006s_e, size_of_lc)

```

Model Recovery 30 Items 6 DIF Symmetric Pattern: 3006s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
```
```

```
# Set DIF type
dif_array = d_b_3006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3006s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3006s_e = rbind(mf_lc3_3006s_e, show(m2))
}

## calculate recovery rate
lcReco_3006s_lc3_e = get_structure_reco_rate(mf_lc3_3006s_e, size_of_lc)
```

Model Recovery 30 Items 12 DIF Symmetric Pattern: 3012s - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_3012s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3012s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3012s_e = rbind(mf_lc3_3012s_e, show(m2))
}
```

```
## calculate recovery rate
lcReco_3012s_lc3_e = get_structure_reco_rate(mf_lc3_3012s_e, size_of_lc)

```

Model Recovery 30 Items 18 DIF Symmetric Pattern: 3018s - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 6
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_3018s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3018s_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3018s_e = rbind(mf_lc3_3018s_e, show(m2))
}

## calculate recovery rate
lcReco_3018s_lc3_e = get_structure_reco_rate(mf_lc3_3018s_e, size_of_lc)

```

Model Recovery - Symmetric Pattern - Three Latent Classes - Unequal Size

Model Recovery 10 Items 2 DIF Symmetric Pattern: 1002s - unequal size
```{r}
set.seed(12345)
```

```
# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1002s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1002s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1002s_u = rbind(mf_lc3_1002s_u, show(m2))
}

## calculate recovery rate
lcReco_1002s_lc3_u = get_structure_reco_rate(mf_lc3_1002s_u, size_of_lc)

```

Model Recovery 10 Items 4 DIF Symmetric Pattern: 1004s - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1004s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1004s_u = list()
```

```
for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1004s_u = rbind(mf_lc3_1004s_u, show(m2))
}

## calculate recovery rate
lcReco_1004s_lc3_u = get_structure_reco_rate(mf_lc3_1004s_u, size_of_lc)

```

Model Recovery 10 Items 6 DIF Symmetric Pattern: 1006s - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1006s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1006s_u = rbind(mf_lc3_1006s_u, show(m2))
}

## calculate recovery rate
lcReco_1006s_lc3_u = get_structure_reco_rate(mf_lc3_1006s_u, size_of_lc)

```

Model Recovery 30 Items 6 DIF Symmetric Pattern: 3006s - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3006s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3006s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3006s_u = rbind(mf_lc3_3006s_u, show(m2))
}

## calculate recovery rate
lcReco_3006s_lc3_u = get_structure_reco_rate(mf_lc3_3006s_u, size_of_lc)

```


Model Recovery 30 Items 12 DIF Symmetric Pattern: 3012s - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 3
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3012s
# Set number of latent classes
```

```
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3012s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3012s_u = rbind(mf_lc3_3012s_u, show(m2))
}

## calculate recovery rate
lcReco_3012s_lc3_u = get_structure_reco_rate(mf_lc3_3012s_u, size_of_lc)
```

Model Recovery 30 Items 18 DIF Symmetric Pattern: 3018s - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 3
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3018s
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3018s_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3018s_u = rbind(mf_lc3_3018s_u, show(m2))
}

## calculate recovery rate
lcReco_3018s_lc3_u = get_structure_reco_rate(mf_lc3_3018s_u, size_of_lc)
```

```

########### Model Recovery - Gradient Pattern - Three Latent Classes - Equal Size ##########

## Model Recovery 10 Items 2 DIF Gradient Pattern: 1002g - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1002g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1002g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1002g_e = rbind(mf_lc3_1002g_e, show(m2))
}

## calculate recovery rate
lcReco_1002g_lc3_e = get_structure_reco_rate(mf_lc3_1002g_e, size_of_lc)

```

Model Recovery 10 Items 4 DIF Gradient Pattern: 1004g - equal size
```{r}
set.seed(12345)

# Set number of replications
```
```

```
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1004g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1004g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1004g_e = rbind(mf_lc3_1004g_e, show(m2))
}

## calculate recovery rate
lcReco_1004g_lc3_e = get_structure_reco_rate(mf_lc3_1004g_e, size_of_lc)
```

Model Recovery 10 Items 6 DIF Gradient Pattern: 1006g - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_1006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1006g_e = list()
```

```
for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1006g_e = rbind(mf_lc3_1006g_e, show(m2))
}

## calculate recovery rate
lcReco_1006g_lc3_e = get_structure_reco_rate(mf_lc3_1006g_e, size_of_lc)

```

Model Recovery 30 Items 6 DIF Gradient Pattern: 3006g - equal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_3006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3006g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3006g_e = rbind(mf_lc3_3006g_e, show(m2))
}

## calculate recovery rate
lcReco_3006g_lc3_e = get_structure_reco_rate(mf_lc3_3006g_e, size_of_lc)

```

Model Recovery 30 Items 12 DIF Gradient Pattern: 3012g - equal size
```{r}
```

```
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_3012g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3012g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3012g_e = rbind(mf_lc3_3012g_e, show(m2))
}

## calculate recovery rate
lcReco_3012g_lc3_e = get_structure_reco_rate(mf_lc3_3012g_e, size_of_lc)
```

Model Recovery 30 Items 18 DIF Gradient Pattern: 3018g - equal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 30
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1000, 1000, 1000)
# Set DIF type
dif_array = d_b_3018g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
```

```
mf_lc3_3018g_e = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3018g_e = rbind(mf_lc3_3018g_e, show(m2))
}

## calculate recovery rate
lcReco_3018g_lc3_e = get_structure_reco_rate(mf_lc3_3018g_e, size_of_lc)

```

######### Model Recovery - Gradient Pattern - Three Latent Classes - Unequal Size #########

Model Recovery 10 Items 2 DIF Gradient Pattern: 1002g - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1002g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1002g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1002g_u = rbind(mf_lc3_1002g_u, show(m2))
}
```

```
## calculate recovery rate
lcReco_1002g_lc3_u = get_structure_reco_rate(mf_lc3_1002g_u, size_of_lc)

```

Model Recovery 10 Items 4 DIF Gradient Pattern: 1004g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1004g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1004g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1004g_u = rbind(mf_lc3_1004g_u, show(m2))
}

## calculate recovery rate
lcReco_1004g_lc3_u = get_structure_reco_rate(mf_lc3_1004g_u, size_of_lc)

```

Model Recovery 10 Items 6 DIF Gradient Pattern: 1006g - unequal size
```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 10
```

```
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_1006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_1006g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_1006g_u = rbind(mf_lc3_1006g_u, show(m2))
}

## calculate recovery rate
lcReco_1006g_lc3_u = get_structure_reco_rate(mf_lc3_1006g_u, size_of_lc)
```

Model Recovery 30 Items 6 DIF Gradient Pattern: 3006g - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3006g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3006g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
```

```
  mf_lc3_3006g_u = rbind(mf_lc3_3006g_u, show(m2))
}

## calculate recovery rate
lcReco_3006g_lc3_u = get_structure_reco_rate(mf_lc3_3006g_u, size_of_lc)
```

Model Recovery 30 Items 12 DIF Gradient Pattern: 3012g - unequal size

```{r}
set.seed(12345)

# Set number of replications
#n_rep = 10
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3012g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3012g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3012g_u = rbind(mf_lc3_3012g_u, show(m2))
}

## calculate recovery rate
lcReco_3012g_lc3_u = get_structure_reco_rate(mf_lc3_3012g_u, size_of_lc)
```

Model Recovery 30 Items 18 DIF Gradient Pattern: 3018g - unequal size

```{r}
set.seed(12345)

# Set number of replications
```

```
#n_rep = 1
# Set number of items
n_item = 30
# Set sample size
size_of_lc = c(1500, 1000, 500)
# Set DIF type
dif_array = d_b_3018g
# Set number of latent classes
n_lc = length(size_of_lc)
# Create list to store model fit data
mf_lc3_3018g_u = list()


for(i in 1:n_rep){
  test = gen_response(n_item, size_of_lc, dif_array)
  m2 = raschmix(as.matrix(test[, 1:n_item]), k = 1:4, scores = "saturated", nrep = 1)
  mf_lc3_3018g_u = rbind(mf_lc3_3018g_u, show(m2))
}

## calculate recovery rate
lcReco_3018g_lc3_u = get_structure_reco_rate(mf_lc3_3018g_u, size_of_lc)
```

```
# Merge Tables and Save to .csv
```{r}
lcReco_lc2_table <- rbind(cbind(lcReco_1002s_lc2_e, lcReco_1002s_lc2_u),
                          cbind(lcReco_1002g_lc2_e, lcReco_1002g_lc2_u),
                          cbind(lcReco_1004s_lc2_e, lcReco_1004s_lc2_u),
                          cbind(lcReco_1004g_lc2_e, lcReco_1004g_lc2_u),
                          cbind(lcReco_1006s_lc2_e, lcReco_1006s_lc2_u),
                          cbind(lcReco_1006g_lc2_e, lcReco_1006g_lc2_u),
                          cbind(lcReco_3006s_lc2_e, lcReco_3006s_lc2_u),
                          cbind(lcReco_3006g_lc2_e, lcReco_3006g_lc2_u),
                          cbind(lcReco_3012s_lc2_e, lcReco_3012s_lc2_u),
                          cbind(lcReco_3012g_lc2_e, lcReco_3012g_lc2_u),
                          cbind(lcReco_3018s_lc2_e, lcReco_3018s_lc2_u),
                          cbind(lcReco_3018g_lc2_e, lcReco_3018g_lc2_u))


lcReco_lc3_table <- rbind(cbind(lcReco_1002s_lc3_e, lcReco_1002s_lc3_u),
```
```

```
                cbind(lcReco_1002g_lc3_e, lcReco_1002g_lc3_u),
                cbind(lcReco_1004s_lc3_e, lcReco_1004s_lc3_u),
                cbind(lcReco_1004g_lc3_e, lcReco_1004g_lc3_u),
                cbind(lcReco_1006s_lc3_e, lcReco_1006s_lc3_u),
                cbind(lcReco_1006g_lc3_e, lcReco_1006g_lc3_u),
                cbind(lcReco_3006s_lc3_e, lcReco_3006s_lc3_u),
                cbind(lcReco_3006g_lc3_e, lcReco_3006g_lc3_u),
                cbind(lcReco_3012s_lc3_e, lcReco_3012s_lc3_u),
                cbind(lcReco_3012g_lc3_e, lcReco_3012g_lc3_u),
                cbind(lcReco_3018s_lc3_e, lcReco_3018s_lc3_u),
                cbind(lcReco_3018g_lc3_e, lcReco_3018g_lc3_u))

```


Generate DIF Descriptive Statistics Table
```{r}
gen_dif_table <- function(df1, df2, df3, df4, df5, df6) {
  table = list()
  m1 = apply(df1, 2, mean)
  sd1 = apply(df1, 2, sd)
  m2 = apply(df2, 2, mean)
  sd2 = apply(df2, 2, sd)
  m3 = apply(df3, 2, mean)
  sd3 = apply(df3, 2, sd)
  m4 = apply(df4, 2, mean)
  sd4 = apply(df4, 2, sd)
  m5 = apply(df5, 2, mean)
  sd5 = apply(df5, 2, sd)
  m6 = apply(df6, 2, mean)
  sd6 = apply(df6, 2, sd)

  table = rbind(m1, sd1, m2, sd2, m3, sd3, m4, sd4, m5, sd5, m6, sd6)
  table = as.matrix(table)
  table = t(table)
  return (table)
}
```

##
######

########################## Parameter Recovery
##########################
##
######

############ Parameter Recovery - Two Latent Classes
##############################

Fit Rasch Mixture Models

######### Parameter Recovery - Equal Size - Two Latent Classes - Symmetric DIF
###########

10 items 2 dif items: 1002S - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002s
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_1002s_lc2_e <- sim[,1:10]
p_1002s_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1002s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1002s, dif_1002s_lc2_e)

# classifier parameter recovery
p_viz(p_1002s_lc2_e, c(1500,1500))
```

10 items 4 dif items: 1004S - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004s
size <- c(1500, 1500)
```

```
sim <- par_reco_sim(dif, size)
dif_1004s_lc2_e <- sim[,1:10]
p_1004s_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1004s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1004s, dif_1004s_lc2_e)

# classifier parameter recovery
p_viz(p_1004s_lc2_e, c(1500, 1500))
```

10 items 6 dif items: 1006S - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006s
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_1006s_lc2_e <- sim[,1:10]
p_1006s_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1006s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1006s, dif_1006s_lc2_e)

# classifier parameter recovery
p_viz(p_1006s_lc2_e, size)
```

30 items 6 dif items: 3006S - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
```

```
dif <- d_b_3006s
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3006s_lc2_e <- sim[,1:30]
p_3006s_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3006s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3006s, dif_3006s_lc2_e)

# classifier parameter recovery
p_viz(p_3006s_lc2_e, size)
```

30 items 12 dif items: 3012S - two latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012s
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3012s_lc2_e <- sim[,1:30]
p_3012s_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3012s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3012s, dif_3012s_lc2_e)

# classifier parameter recovery
p_viz(p_3012s_lc2_e, size)
```

30 items 18 dif items: 3018S - two latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018s
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3018s_lc2_e <- sim[,1:30]
p_3018s_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3018s_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3018s, dif_3018s_lc2_e)

# classifier parameter recovery
p_viz(p_3018s_lc2_e, size)
```

Parameter Recovery - Unequal Size - Two Latent Classes - Symmetric DIF#############

10 items 2 dif items: 1002S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1002s_lc2_u <- sim[,1:10]
p_1002s_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1002s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1002s, dif_1002s_lc2_u)
```

```
# classifier parameter recovery
p_viz(p_1002s_lc2_u, size)
```

10 items 4 dif items: 1004S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1004s_lc2_u <- sim[,1:10]
p_1004s_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1004s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1004s, dif_1004s_lc2_u)

# classifier parameter recovery
p_viz(p_1004s_lc2_u, size)
```

10 items 6 dif items: 1006S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1006s_lc2_u <- sim[,1:10]
p_1006s_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1006s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
```

```
# dif recovery
dif_viz(d_b_1006s, dif_1006s_lc2_u)

# classifier parameter recovery
p_viz(p_1006s_lc2_u, size)
```

30 items 6 dif items: 3006S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3006s_lc2_u <- sim[,1:30]
p_3006s_lc2_u <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3006s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3006s, dif_3006s_lc2_u)

# classifier parameter recovery
p_viz(p_3006s_lc2_u, size)

```

30 items 12 dif items: 3012S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3012s_lc2_u <- sim[,1:30]
p_3012s_lc2_u <- sim[,31:32]

# Stop the clock and sotre runtime in rt
```

```
rt_3012s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3012s, dif_3012s_lc2_u)

# classifier parameter recovery
p_viz(p_3012s_lc2_u, size)
```

30 items 18 dif items: 3018S - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018s
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3018s_lc2_u <- sim[,1:30]
p_3018s_lc2_u <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3018s_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3018s, dif_3018s_lc2_u)

# classifier parameter recovery
p_viz(p_3018s_lc2_u, size)
```

########## Parameter Recovery - Equal Size - Two Latent Classes - Gradient DIF ############

10 items 2 dif items: 1002g - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002g
size <- c(1500, 1500)
```

```
sim <- par_reco_sim(dif, size)
dif_1002g_lc2_e <- sim[,1:10]
p_1002g_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1002g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1002g, dif_1002g_lc2_e)

# classifier parameter recovery
p_viz(p_1002g_lc2_e, size)
```

10 items 4 dif items: 1004g - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004g
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_1004g_lc2_e <- sim[,1:10]
p_1004g_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1004g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1004g, dif_1004g_lc2_e)

# classifier parameter recovery
p_viz(p_1004g_lc2_e, size)
```

10 items 6 dif items: 1006g - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
```

```
dif <- d_b_1006g
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_1006g_lc2_e <- sim[,1:10]
p_1006g_lc2_e <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1006g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1006g, dif_1006g_lc2_e)

# classifier parameter recovery
p_viz(p_1006g_lc2_e, size)
```

30 items 6 dif items: 3006g - two latent classes - equal size design
```{r, warning = False}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006g
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3006g_lc2_e <- sim[,1:30]
p_3006g_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3006g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3006g, dif_3006g_lc2_e)

# classifier parameter recovery
p_viz(p_3006g_lc2_e, size)
```

30 items 12 dif items: 3012g - two latent classes - equal size design
```{r}
```

```
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012g
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3012g_lc2_e <- sim[,1:30]
p_3012g_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3012g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3012g, dif_3012g_lc2_e)

# classifier parameter recovery
p_viz(p_3012g_lc2_e, size)
```

30 items 18 dif items: 3018g - two latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018g
size <- c(1500, 1500)

sim <- par_reco_sim(dif, size)
dif_3018g_lc2_e <- sim[,1:30]
p_3018g_lc2_e <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3018g_lc2_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3018g, dif_3018g_lc2_e)

# classifier parameter recovery
p_viz(p_3018g_lc2_e, size)
```

Parameter Recovery - Unequal Size - Two Latent Classes - Gradient DIF

10 items 2 dif items: 1002g - two latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1002g_lc2_u <- sim[,1:10]
p_1002g_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1002g_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1002g, dif_1002g_lc2_u)

# classifier parameter recovery
p_viz(p_1002g_lc2_u, size)
```

10 items 4 dif items: 1004g - two latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1004g_lc2_u <- sim[,1:10]
p_1004g_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1004g_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
```

```
dif_viz(d_b_1004g, dif_1004g_lc2_u)

# classifier parameter recovery
p_viz(p_1004g_lc2_u, size)
```

10 items 6 dif items: 1006g - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_1006g_lc2_u <- sim[,1:10]
p_1006g_lc2_u <- sim[,11:12]

# Stop the clock and sotre runtime in rt
rt_1006g_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_1006g, dif_1006g_lc2_u)

# classifier parameter recovery
p_viz(p_1006g_lc2_u, size)
```

30 items 6 dif items: 3006g - two latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3006g_lc2_u <- sim[,1:30]
p_3006g_lc2_u <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3006g_lc2_u = (proc.time() - ptm)[1:3]
```

```
# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3006g, dif_3006g_lc2_u)

# classifier parameter recovery
p_viz(p_3006g_lc2_u, size)
```

30 items 12 dif items: 3012g - two latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3012g_lc2_u <- sim[,1:30]
p_3012g_lc2_u <- sim[,31:32]

# Stop the clock and sotre runtime in rt
rt_3012g_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3012g, dif_3012g_lc2_u)

# classifier parameter recovery
p_viz(p_3012g_lc2_u, size)
```

30 items 18 dif items: 3018g - two latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018g
size <- c(2000, 1000)

sim <- par_reco_sim(dif, size)
dif_3018g_lc2_u <- sim[,1:30]
p_3018g_lc2_u <- sim[,31:32]
```

```
# Stop the clock and sotre runtime in rt
rt_3018g_lc2_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(d_b_3018g, dif_3018g_lc2_u)

# classifier parameter recovery
p_viz(p_3018g_lc2_u, size)
```


################## Parameter Recovery - Three Latent Classes
########################

Fit Rasch Mixture Models

######### Parameter Recovery - Equal Size - Three Latent Classes - Symmetric DIF
##########

10 items 2 dif items: 1002S - three latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1002s_lc3_e_1 <- sim[, 1:10]
dif_1002s_lc3_e_2 <- sim[, 11:20]
p_1002s_lc3_e <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1002s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1002s_lc3_e_1)
dif_viz(dif_2, dif_1002s_lc3_e_2)

# classifier parameter recovery
```

```
p_viz(p_1002s_lc3_e, size)

```

10 items 4 dif items: 1004S - three latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1004s_lc3_e_1 <- sim[, 1:10]
dif_1004s_lc3_e_2 <- sim[, 11:20]
p_1004s_lc3_e <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1004s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1004s_lc3_e_1)
dif_viz(dif_2, dif_1004s_lc3_e_2)

# classifier parameter recovery
p_viz(p_1004s_lc3_e, size)

```

10 items 6 dif items: 1006S - three latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1006s_lc3_e_1 <- sim[,1:10]
dif_1006s_lc3_e_2 <- sim[,11:20]
p_1006s_lc3_e <- sim[,21:23]
```

```
# Stop the clock and sotre runtime in rt
rt_1006s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1006s_lc3_e_1)
dif_viz(dif_2, dif_1006s_lc3_e_2)

# classifier parameter recovery
p_viz(p_1006s_lc3_e, size)
```

30 items 6 dif items: 3006S - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3006s_lc3_e_1 <- sim[,1:30]
dif_3006s_lc3_e_2 <- sim[,31:60]
p_3006s_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3006s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3006s_lc3_e_1)
dif_viz(dif_2, dif_3006s_lc3_e_2)

# classifier parameter recovery
p_viz(p_3006s_lc3_e, size)
```

30 items 12 dif items: 3012S - three latent classes - equal size design

```{r}
```

```
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3012s_lc3_e_1 <- sim[,1:30]
dif_3012s_lc3_e_2 <- sim[,31:60]
p_3012s_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3012s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3012s_lc3_e_1)
dif_viz(dif_2, dif_3012s_lc3_e_2)

# classifier parameter recovery
p_viz(p_3012s_lc3_e, size)
```

30 items 18 dif items: 3018S - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018s
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3018s_lc3_e_1 <- sim[,1:30]
dif_3018s_lc3_e_2 <- sim[,31:60]
p_3018s_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3018s_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
```

```
dif_viz(dif, dif_3018s_lc3_e_1)
dif_viz(dif_2, dif_3018s_lc3_e_2)

# classifier parameter recovery
p_viz(p_3018s_lc3_e, size)

```


######### Parameter Recovery - Unequal Size - Three Latent Classes - Symmetric DIF ########

10 items 2 dif items: 1002S - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1002s_lc3_u_1 <- sim[,1:10]
dif_1002s_lc3_u_2 <- sim[,11:20]
p_1002s_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1002s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1002s_lc3_u_1)
dif_viz(dif_2, dif_1002s_lc3_u_2)

# classifier parameter recovery
p_viz(p_1002s_lc3_u, size)

```


10 items 4 dif items: 1004S - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
```

```
dif <- d_b_1004s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1004s_lc3_u_1 <- sim[,1:10]
dif_1004s_lc3_u_2 <- sim[,11:20]
p_1004s_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1004s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1004s_lc3_u_1)
dif_viz(dif_2, dif_1004s_lc3_u_2)

# classifier parameter recovery
p_viz(p_1004s_lc3_u, size)
```

10 items 6 dif items: 1006S - three latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1006s_lc3_u_1 <- sim[,1:10]
dif_1006s_lc3_u_2 <- sim[,11:20]
p_1006s_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1006s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1006s_lc3_u_1)
dif_viz(dif_2, dif_1006s_lc3_u_2)
```

```
# classifier parameter recovery
p_viz(p_1006s_lc3_u, size)

```

30 items 6 dif items: 3006S - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3006s_lc3_u_1 <- sim[,1:30]
dif_3006s_lc3_u_2 <- sim[,31:60]
p_3006s_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3006s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3006s_lc3_u_1)
dif_viz(dif_2, dif_3006s_lc3_u_2)

# classifier parameter recovery
p_viz(p_3006s_lc3_u, size)

```

30 items 12 dif items: 3012S - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3012s_lc3_u_1 <- sim[,1:30]
```

```
dif_3012s_lc3_u_2 <- sim[,31:60]
p_3012s_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3012s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3012s_lc3_u_1)
dif_viz(dif_2, dif_3012s_lc3_u_2)

# classifier parameter recovery
p_viz(p_3012s_lc3_u, size)
```

30 items 18 dif items: 3018S - three latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018s
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3018s_lc3_u_1 <- sim[,1:30]
dif_3018s_lc3_u_2 <- sim[,31:60]
p_3018s_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3018s_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3018s_lc3_u_1)
dif_viz(dif_2, dif_3018s_lc3_u_2)

# classifier parameter recovery
p_viz(p_3018s_lc3_u, size)
```

########### Parameter Recovery - Equal Size - Three Latent Classes - Gradient DIF ##########

10 items 2 dif items: 1002g - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1002g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1002g_lc3_e_1 <- sim[,1:10]
dif_1002g_lc3_e_2 <- sim[,11:20]
p_1002g_lc3_e <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1002g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1002g_lc3_e_1)
dif_viz(dif_2, dif_1002g_lc3_e_2)

# classifier parameter recovery
p_viz(p_1002g_lc3_e, size)

```

10 items 4 dif items: 1004g - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1004g_lc3_e_1 <- sim[,1:10]
dif_1004g_lc3_e_2 <- sim[,11:20]
p_1004g_lc3_e <- sim[,21:23]
```

```
# Stop the clock and sotre runtime in rt
rt_1004g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1004g_lc3_e_1)
dif_viz(dif_2, dif_1004g_lc3_e_2)

# classifier parameter recovery
p_viz(p_1004g_lc3_e, size)
```

10 items 6 dif items: 1006g - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_1006g_lc3_e_1 <- sim[,1:10]
dif_1006g_lc3_e_2 <- sim[,11:20]
p_1006g_lc3_e <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1006g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1006g_lc3_e_1)
dif_viz(dif_2, dif_1006g_lc3_e_2)

# classifier parameter recovery
p_viz(p_1006g_lc3_e, size)
```

30 items 6 dif items: 3006g - three latent classes - equal size design

```{r, warning = False}
```

```
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3006g_lc3_e_1 <- sim[,1:30]
dif_3006g_lc3_e_2 <- sim[,31:60]
p_3006g_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3006g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3006g_lc3_e_1)
dif_viz(dif_2, dif_3006g_lc3_e_2)

# classifier parameter recovery
p_viz(p_3006g_lc3_e, size)
```

30 items 12 dif items: 3012g - three latent classes - equal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3012g_lc3_e_1 <- sim[,1:30]
dif_3012g_lc3_e_2 <- sim[,31:60]
p_3012g_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3012g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
```

```
dif_viz(dif, dif_3012g_lc3_e_1)
dif_viz(dif_2, dif_3012g_lc3_e_2)

# classifier parameter recovery
p_viz(p_3012g_lc3_e, size)

```

30 items 18 dif items: 3018g - three latent classes - equal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018g
dif_2 <- dif*2
size <- c(1000, 1000, 1000)

sim <- par_reco_sim(dif, size)
dif_3018g_lc3_e_1 <- sim[,1:30]
dif_3018g_lc3_e_2 <- sim[,31:60]
p_3018g_lc3_e <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3018g_lc3_e = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3018g_lc3_e_1)
dif_viz(dif_2, dif_3018g_lc3_e_2)

# classifier parameter recovery
p_viz(p_3018g_lc3_e, size)

```


######## Parameter Recovery - Unequal Size - Three Latent Classes - Gradient DIF ##########

10 items 2 dif items: 1002g - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
```

```
dif <- d_b_1002g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1002g_lc3_u_1 <- sim[,1:10]
dif_1002g_lc3_u_2 <- sim[,11:20]
p_1002g_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1002g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1002g_lc3_u_1)
dif_viz(dif_2, dif_1002g_lc3_u_2)

# classifier parameter recovery
p_viz(p_1002g_lc3_u, size)
```

10 items 4 dif items: 1004g - three latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1004g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1004g_lc3_u_1 <- sim[,1:10]
dif_1004g_lc3_u_2 <- sim[,11:20]
p_1004g_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1004g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1004g_lc3_u_1)
dif_viz(dif_2, dif_1004g_lc3_u_2)
```

```
# classifier parameter recovery
p_viz(p_1004g_lc3_u, size)

```

10 items 6 dif items: 1006g - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_1006g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_1006g_lc3_u_1 <- sim[,1:10]
dif_1006g_lc3_u_2 <- sim[,11:20]
p_1006g_lc3_u <- sim[,21:23]

# Stop the clock and sotre runtime in rt
rt_1006g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_1006g_lc3_u_1)
dif_viz(dif_2, dif_1006g_lc3_u_2)

# classifier parameter recovery
p_viz(p_1006g_lc3_u, size)

```

30 items 6 dif items: 3006g - three latent classes - unequal size design
```{r, warning = False}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3006g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3006g_lc3_u_1 <- sim[,1:30]
```

```
dif_3006g_lc3_u_2 <- sim[,31:60]
p_3006g_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3006g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3006g_lc3_u_1)
dif_viz(dif_2, dif_3006g_lc3_u_2)

# classifier parameter recovery
p_viz(p_3006g_lc3_u, size)
```

30 items 12 dif items: 3012g - three latent classes - unequal size design

```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3012g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3012g_lc3_u_1 <- sim[,1:30]
dif_3012g_lc3_u_2 <- sim[,31:60]
p_3012g_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3012g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3012g_lc3_u_1)
dif_viz(dif_2, dif_3012g_lc3_u_2)

# classifier parameter recovery
p_viz(p_3012g_lc3_u, size)
```

30 items 18 dif items: 3018g - three latent classes - unequal size design
```{r}
# Add time recorder: start the clock
ptm <- proc.time()
dif <- d_b_3018g
dif_2 <- dif*2
size <- c(1500, 1000, 500)

sim <- par_reco_sim(dif, size)
dif_3018g_lc3_u_1 <- sim[,1:30]
dif_3018g_lc3_u_2 <- sim[,31:60]
p_3018g_lc3_u <- sim[,61:63]

# Stop the clock and sotre runtime in rt
rt_3018g_lc3_u = (proc.time() - ptm)[1:3]

# Visualize parameter recovery
# dif recovery
dif_viz(dif, dif_3018g_lc3_u_1)
dif_viz(dif_2, dif_3018g_lc3_u_2)

# classifier parameter recovery
p_viz(p_3018g_lc3_u, size)

```


Create descriptive statistics table for each test type
```{r}
table_1002s = gen_dif_table(dif_1002s_lc2_e, dif_1002s_lc2_u, dif_1002s_lc3_e_1,
dif_1002s_lc3_u_1, dif_1002s_lc3_e_2, dif_1002s_lc3_u_2)

table_1004s = gen_dif_table(dif_1004s_lc2_e, dif_1004s_lc2_u, dif_1004s_lc3_e_1,
dif_1004s_lc3_u_1, dif_1004s_lc3_e_2, dif_1004s_lc3_u_2)

table_1006s = gen_dif_table(dif_1006s_lc2_e, dif_1006s_lc2_u, dif_1006s_lc3_e_1,
dif_1006s_lc3_u_1, dif_1006s_lc3_e_2, dif_1006s_lc3_u_2)

table_3006s = gen_dif_table(dif_3006s_lc2_e, dif_3006s_lc2_u, dif_3006s_lc3_e_1,
dif_3006s_lc3_u_1, dif_3006s_lc3_e_2, dif_3006s_lc3_u_2)
```

```
table_3012s = gen_dif_table(dif_3012s_lc2_e, dif_3012s_lc2_u, dif_3012s_lc3_e_1,
dif_3012s_lc3_u_1, dif_3012s_lc3_e_2, dif_3012s_lc3_u_2)

table_3018s = gen_dif_table(dif_3018s_lc2_e, dif_3018s_lc2_u, dif_3018s_lc3_e_1,
dif_3018s_lc3_u_1, dif_3018s_lc3_e_2, dif_3018s_lc3_u_2)

table_1002g = gen_dif_table(dif_1002g_lc2_e, dif_1002g_lc2_u, dif_1002g_lc3_e_1,
dif_1002g_lc3_u_1, dif_1002g_lc3_e_2, dif_1002g_lc3_u_2)

table_1004g = gen_dif_table(dif_1004g_lc2_e, dif_1004g_lc2_u, dif_1004g_lc3_e_1,
dif_1004g_lc3_u_1, dif_1004g_lc3_e_2, dif_1004g_lc3_u_2)

table_1006g = gen_dif_table(dif_1006g_lc2_e, dif_1006g_lc2_u, dif_1006g_lc3_e_1,
dif_1006g_lc3_u_1, dif_1006g_lc3_e_2, dif_1006g_lc3_u_2)

table_3006g = gen_dif_table(dif_3006g_lc2_e, dif_3006g_lc2_u, dif_3006g_lc3_e_1,
dif_3006g_lc3_u_1, dif_3006g_lc3_e_2, dif_3006g_lc3_u_2)

table_3012g = gen_dif_table(dif_3012g_lc2_e, dif_3012g_lc2_u, dif_3012g_lc3_e_1,
dif_3012g_lc3_u_1, dif_3012g_lc3_e_2, dif_3012g_lc3_u_2)

table_3018g = gen_dif_table(dif_3018g_lc2_e, dif_3018g_lc2_u, dif_3018g_lc3_e_1,
dif_3018g_lc3_u_1, dif_3018g_lc3_e_2, dif_3018g_lc3_u_2)

# Write tables to csv files for future editing
write.csv(table_1002s, "table_1002s.csv")
write.csv(table_1004s, "table_1004s.csv")
write.csv(table_1006s, "table_1006s.csv")

write.csv(table_1002g, "table_1002g.csv")
write.csv(table_1004g, "table_1004g.csv")
write.csv(table_1006g, "table_1006g.csv")

write.csv(table_3006s, "table_3006s.csv")
write.csv(table_3012s, "table_3012s.csv")
write.csv(table_3018s, "table_3018s.csv")

write.csv(table_3006g, "table_3006g.csv")
write.csv(table_3012g, "table_3012g.csv")
write.csv(table_3018g, "table_3018g.csv")

```

Output classifier parameter recovery plots in a combined way.
```{r}
par(mfrow=c(2, 2))
p_viz(p_1002s_lc2_e, c(1500, 1500))
p_viz(p_1002s_lc2_u, c(2000, 1000))
p_viz(p_1002s_lc3_e, c(1000, 1000, 1000))
p_viz(p_1002s_lc3_u, c(1500, 1000, 500))

p_viz(p_1002g_lc2_e, c(1500, 1500))
p_viz(p_1002g_lc2_u, c(2000, 1000))
p_viz(p_1002g_lc3_e, c(1000, 1000, 1000))
p_viz(p_1002g_lc3_u, c(1500, 1000, 500))

p_viz(p_1004s_lc2_e, c(1500, 1500))
p_viz(p_1004s_lc2_u, c(2000, 1000))
p_viz(p_1004s_lc3_e, c(1000, 1000, 1000))
p_viz(p_1004s_lc3_u, c(1500, 1000, 500))

p_viz(p_1004g_lc2_e, c(1500, 1500))
p_viz(p_1004g_lc2_u, c(2000, 1000))
p_viz(p_1004g_lc3_e, c(1000, 1000, 1000))
p_viz(p_1004g_lc3_u, c(1500, 1000, 500))

p_viz(p_1006s_lc2_e, c(1500, 1500))
p_viz(p_1006s_lc2_u, c(2000, 1000))
p_viz(p_1006s_lc3_e, c(1000, 1000, 1000))
p_viz(p_1006s_lc3_u, c(1500, 1000, 500))

p_viz(p_1006g_lc2_e, c(1500, 1500))
p_viz(p_1006g_lc2_u, c(2000, 1000))
p_viz(p_1006g_lc3_e, c(1000, 1000, 1000))
p_viz(p_1006g_lc3_u, c(1500, 1000, 500))

p_viz(p_3006s_lc2_e, c(1500, 1500))
p_viz(p_3006s_lc2_u, c(2000, 1000))
p_viz(p_3006s_lc3_e, c(1000, 1000, 1000))
p_viz(p_3006s_lc3_u, c(1500, 1000, 500))

p_viz(p_3006g_lc2_e, c(1500, 1500))
p_viz(p_3006g_lc2_u, c(2000, 1000))
```

```
p_viz(p_3006g_lc3_e, c(1000, 1000, 1000))
p_viz(p_3006g_lc3_u, c(1500, 1000, 500))

p_viz(p_3012s_lc2_e, c(1500, 1500))
p_viz(p_3012s_lc2_u, c(2000, 1000))
p_viz(p_3012s_lc3_e, c(1000, 1000, 1000))
p_viz(p_3012s_lc3_u, c(1500, 1000, 500))

p_viz(p_3012g_lc2_e, c(1500, 1500))
p_viz(p_3012g_lc2_u, c(2000, 1000))
p_viz(p_3012g_lc3_e, c(1000, 1000, 1000))
p_viz(p_3012g_lc3_u, c(1500, 1000, 500))

p_viz(p_3018s_lc2_e, c(1500, 1500))
p_viz(p_3018s_lc2_u, c(2000, 1000))
p_viz(p_3018s_lc3_e, c(1000, 1000, 1000))
p_viz(p_3018s_lc3_u, c(1500, 1000, 500))

p_viz(p_3018g_lc2_e, c(1500, 1500))
p_viz(p_3018g_lc2_u, c(2000, 1000))
p_viz(p_3018g_lc3_e, c(1000, 1000, 1000))
p_viz(p_3018g_lc3_u, c(1500, 1000, 500))

```
```{r}
rt_table <- rbind(rt_1002s_lc2_e, rt_1002s_lc2_u, rt_1002s_lc3_e, rt_1002s_lc3_u,
                  rt_1002g_lc2_e, rt_1002g_lc2_u, rt_1002g_lc3_e, rt_1002g_lc3_u,
                  rt_1004s_lc2_e, rt_1004s_lc2_u, rt_1004s_lc3_e, rt_1004s_lc3_u,
                  rt_1004g_lc2_e, rt_1004g_lc2_u, rt_1004g_lc3_e, rt_1004g_lc3_u,
                  rt_1006s_lc2_e, rt_1006s_lc2_u, rt_1006s_lc3_e, rt_1006s_lc3_u,
                  rt_1006g_lc2_e, rt_1006g_lc2_u, rt_1006g_lc3_e, rt_1006g_lc3_u,
                  rt_3006s_lc2_e, rt_3006s_lc2_u, rt_3006s_lc3_e, rt_3006s_lc3_u,
                  rt_3006g_lc2_e, rt_3006g_lc2_u, rt_3006g_lc3_e, rt_3006g_lc3_u,
                  rt_3012s_lc2_e, rt_3012s_lc2_u, rt_3012s_lc3_e, rt_3012s_lc3_u,
                  rt_3012g_lc2_e, rt_3012g_lc2_u, rt_3012g_lc3_e, rt_3012g_lc3_u,
                  rt_3018s_lc2_e, rt_3018s_lc2_u, rt_3018s_lc3_e, rt_3018s_lc3_u,
                  rt_3018g_lc2_e, rt_3018g_lc2_u, rt_3018g_lc3_e, rt_3018g_lc3_u)
write.csv(rt_table, "Runtime Table.csv")
```

Appendix B Figures for Item Level DIF Recovery

Figure 1
1002s_lc2_e Item DIF Recovery

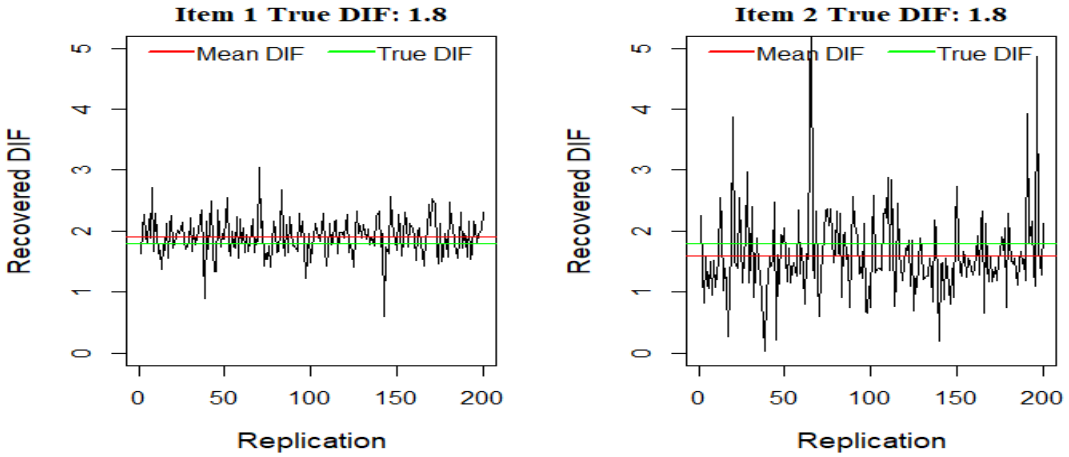

Figure 2
1004 lc2 e Item DIF Recovery

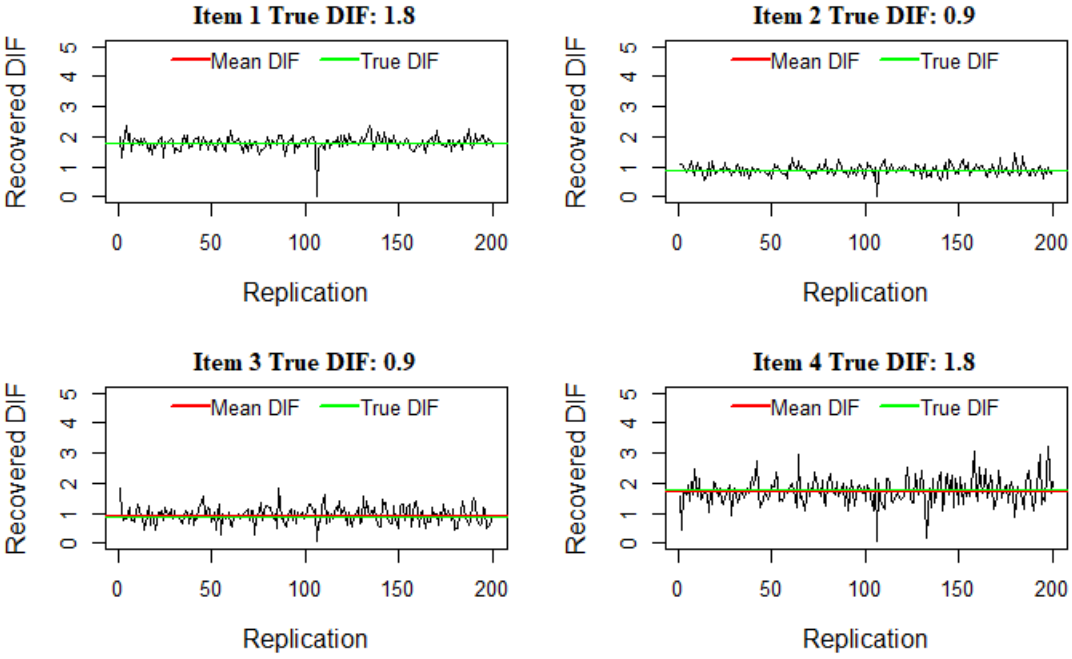

Figure 3
1006s_lc2_e Item DIF Recovery

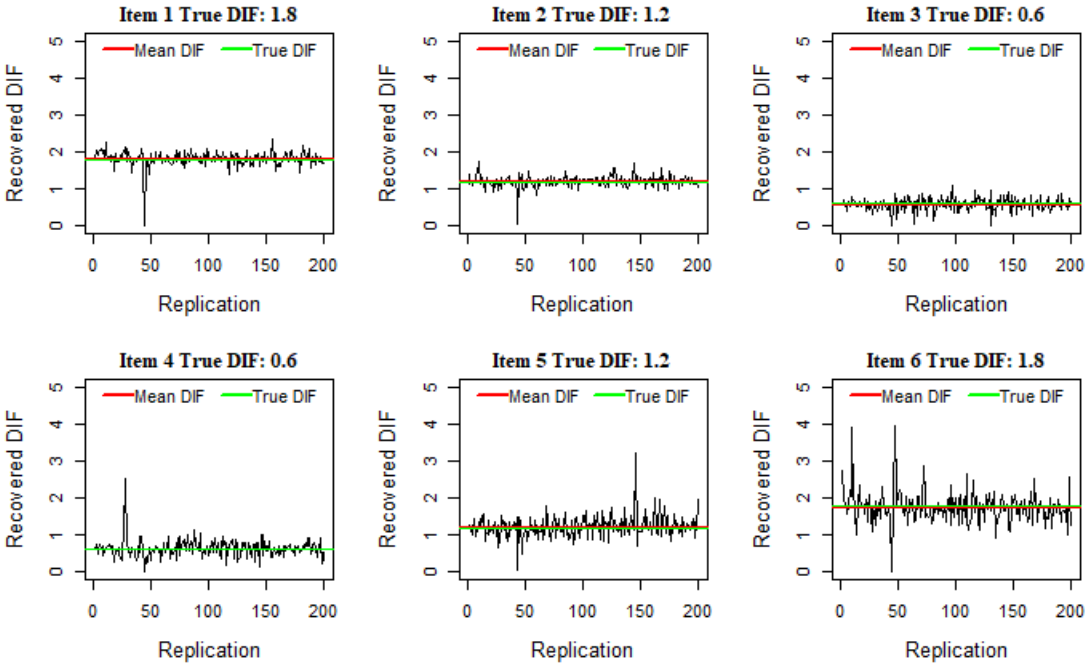

Figure 4
3006s_lc2_e Item DIF Recovery

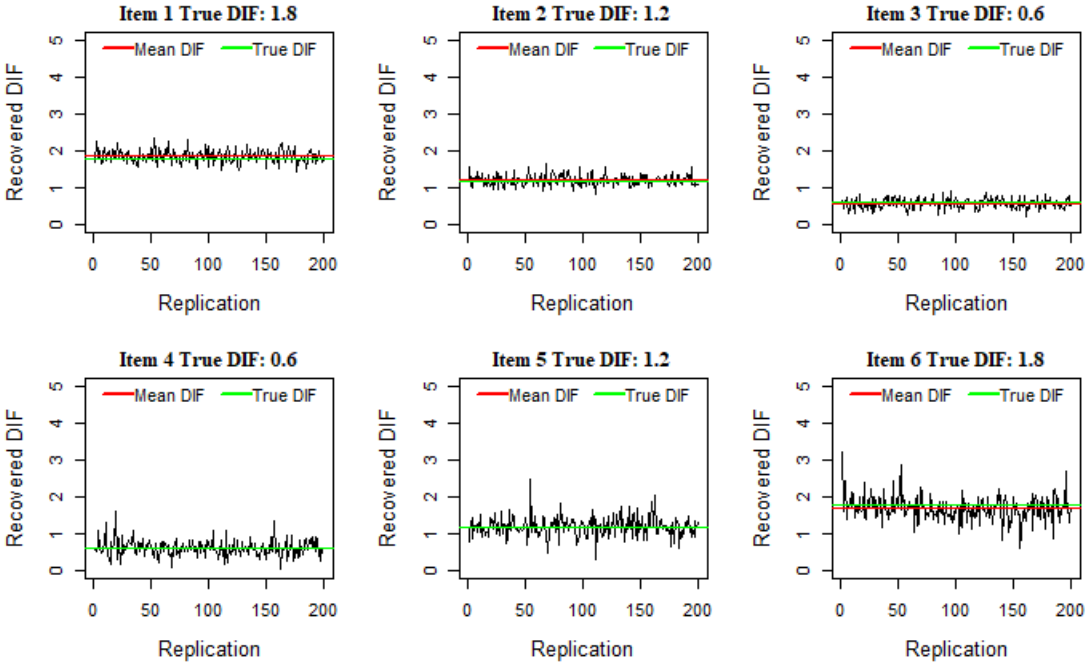

Figure 5

3012s_lc2_e Item DIF Recovery

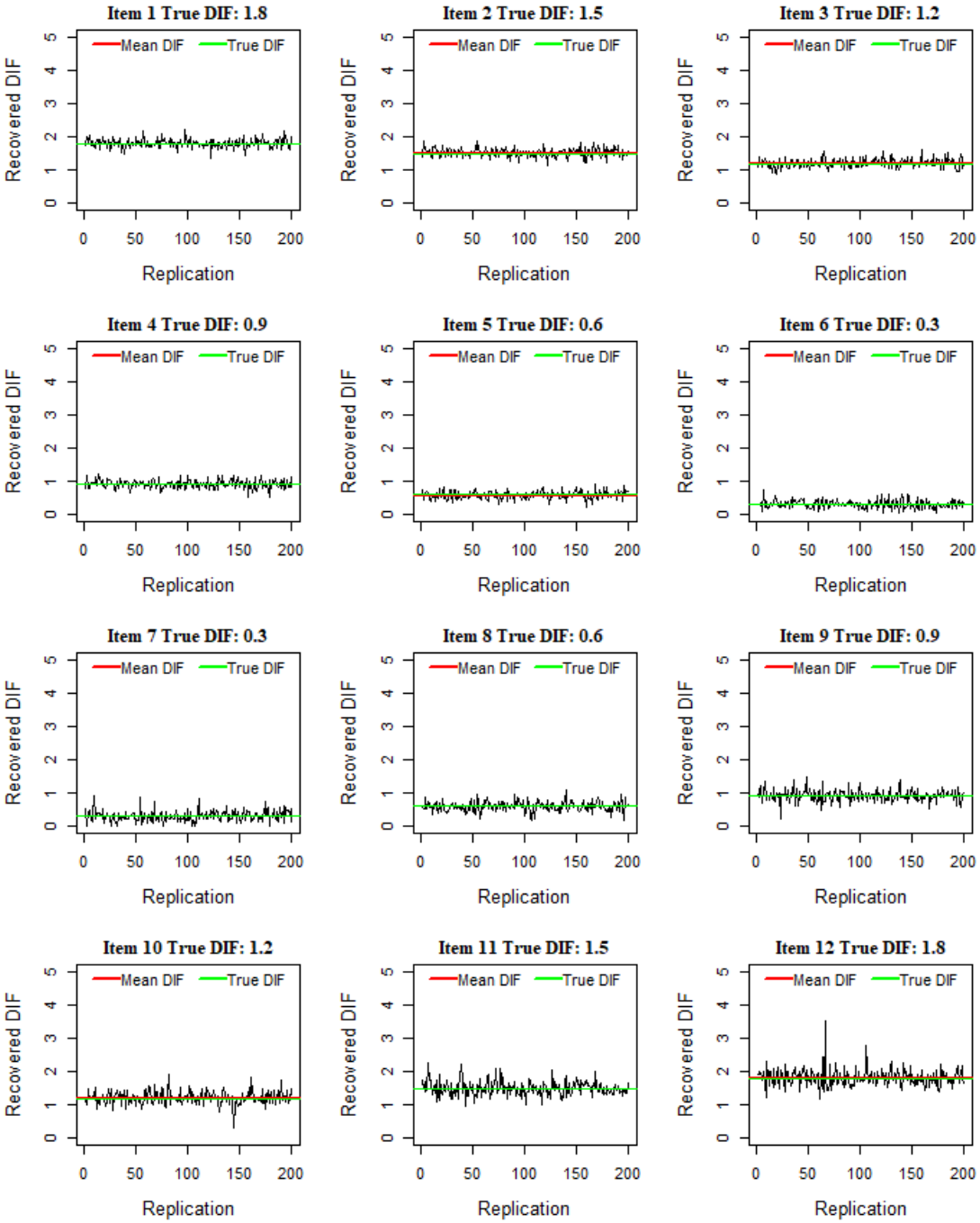

Figure 6
3018s_lc2_e Item DIF Recovery

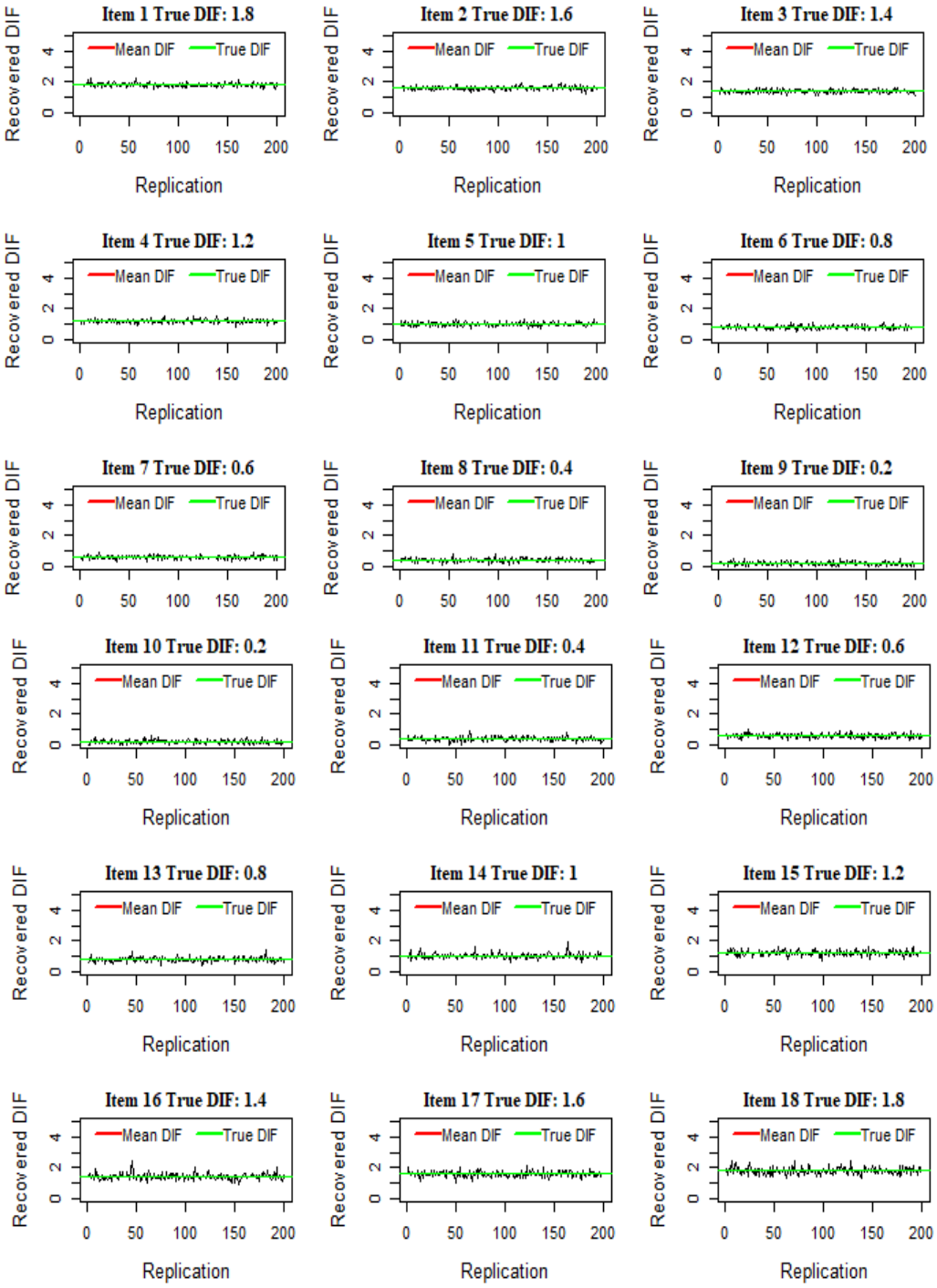

Figure 7
1002s_lc2_u Item DIF Recovery

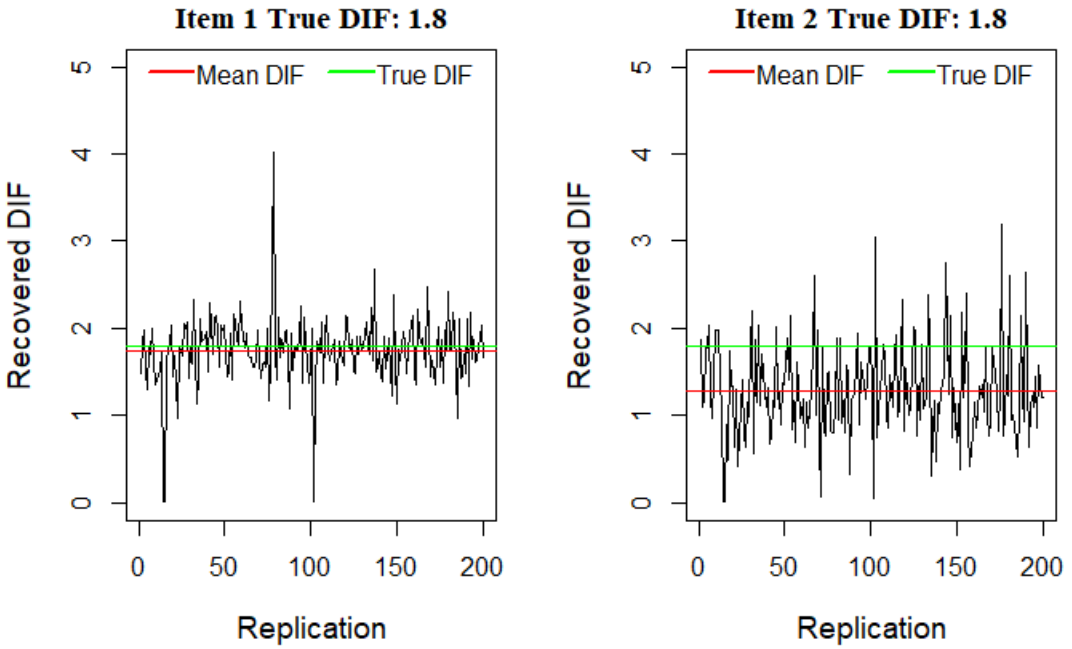

Figure 8
1004s_lc2_u Item DIF Recovery

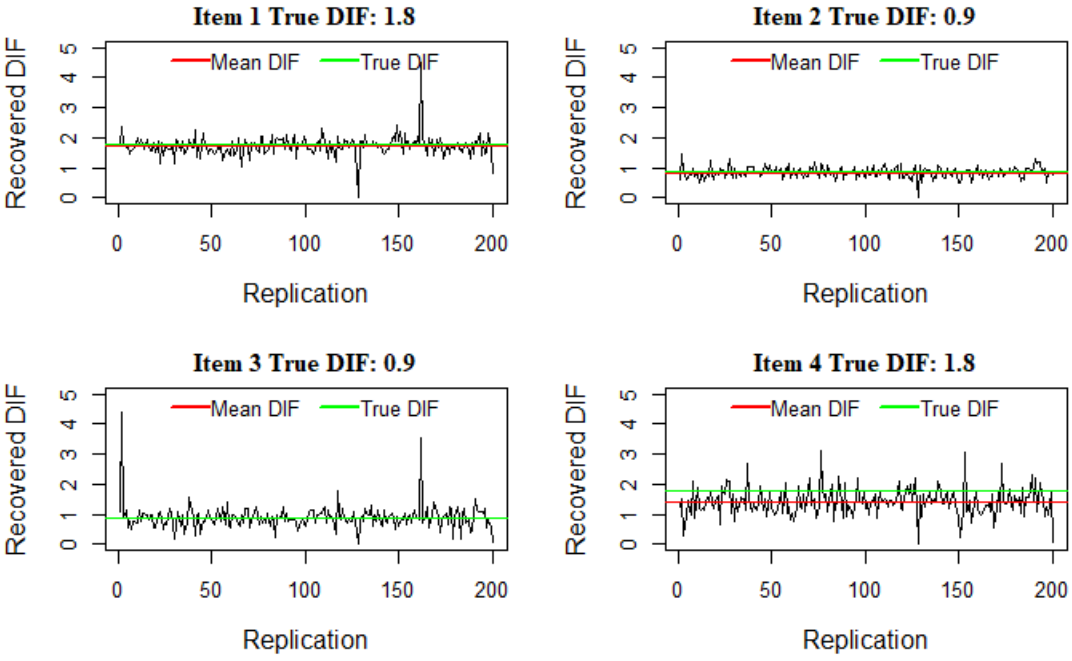

Figure 9
1006s_lc2_u Item DIF Recovery

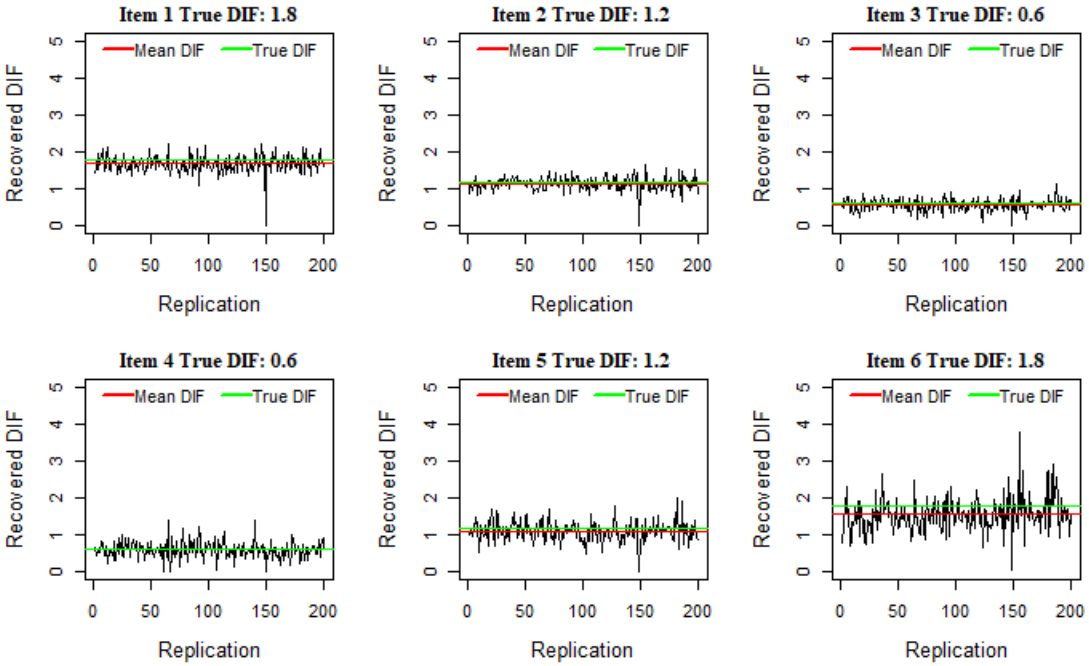

Figure 10
3006s_lc2_u Item DIF Recovery

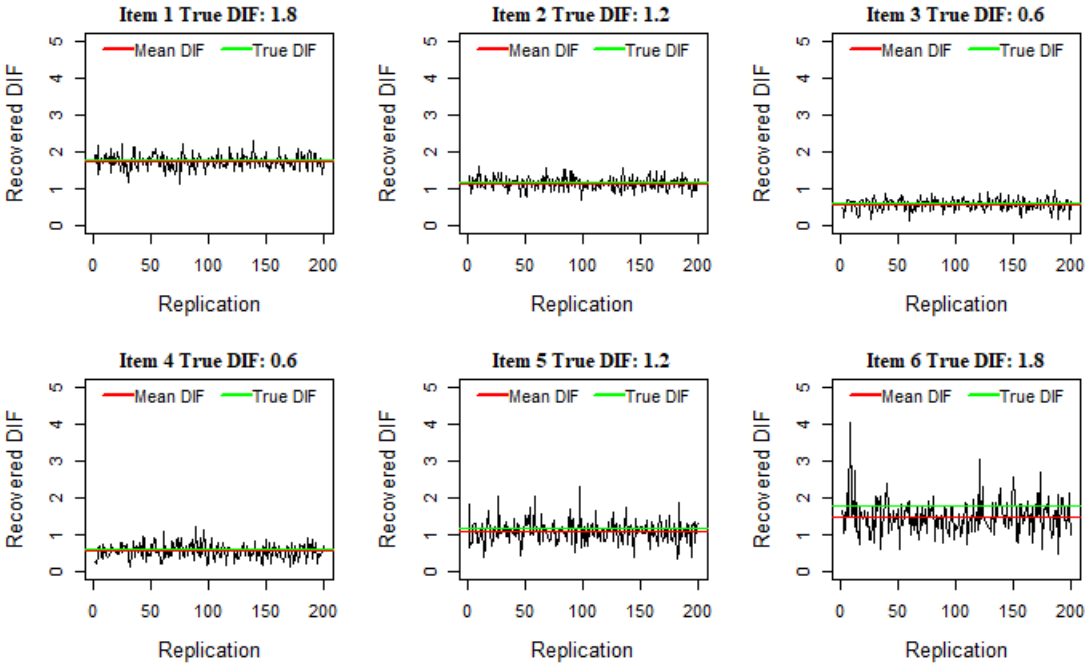

Figure 11
3012s_lc2_u Item DIF Recovery

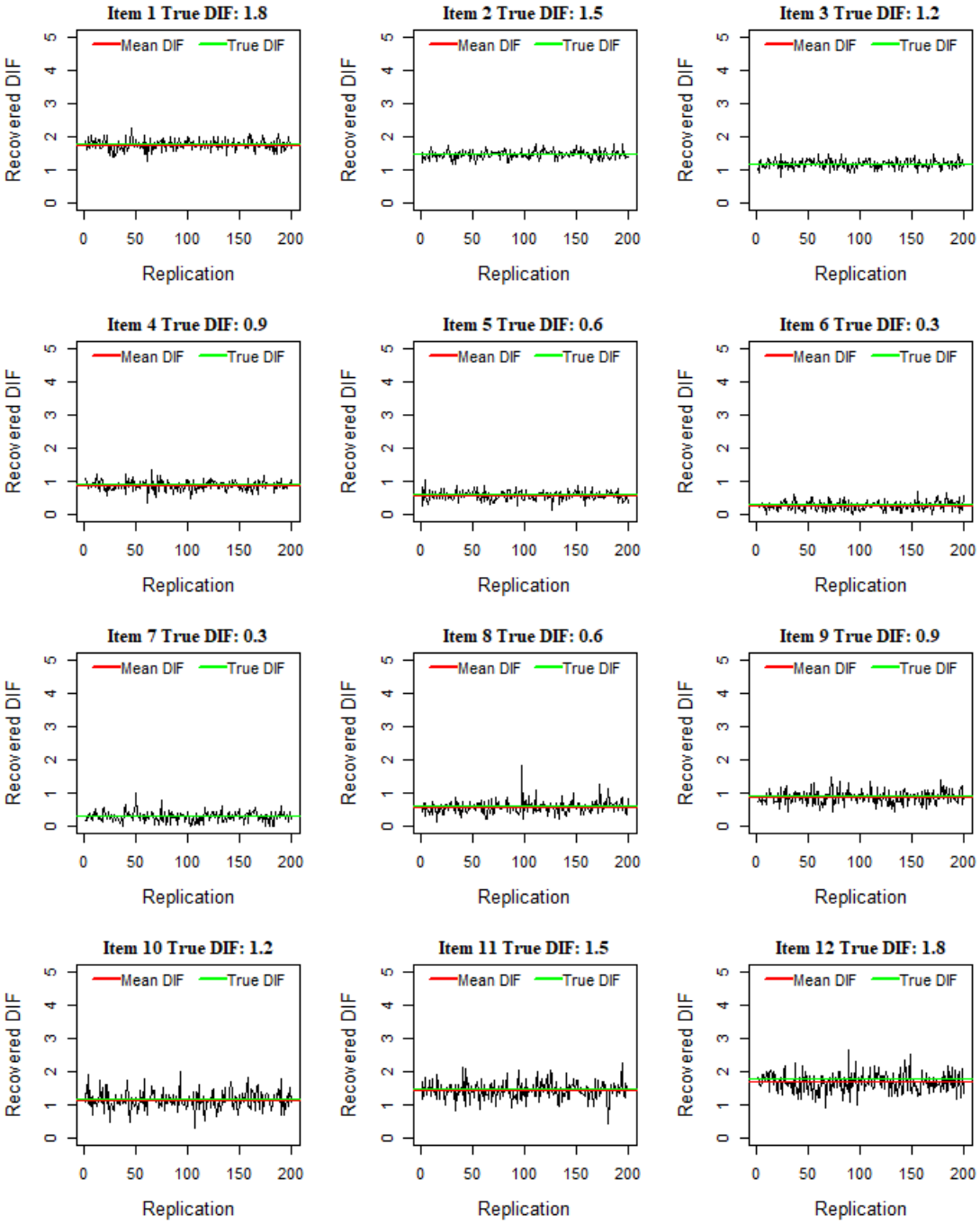

Figure 12
3018s_lc2_u Item DIF Recovery

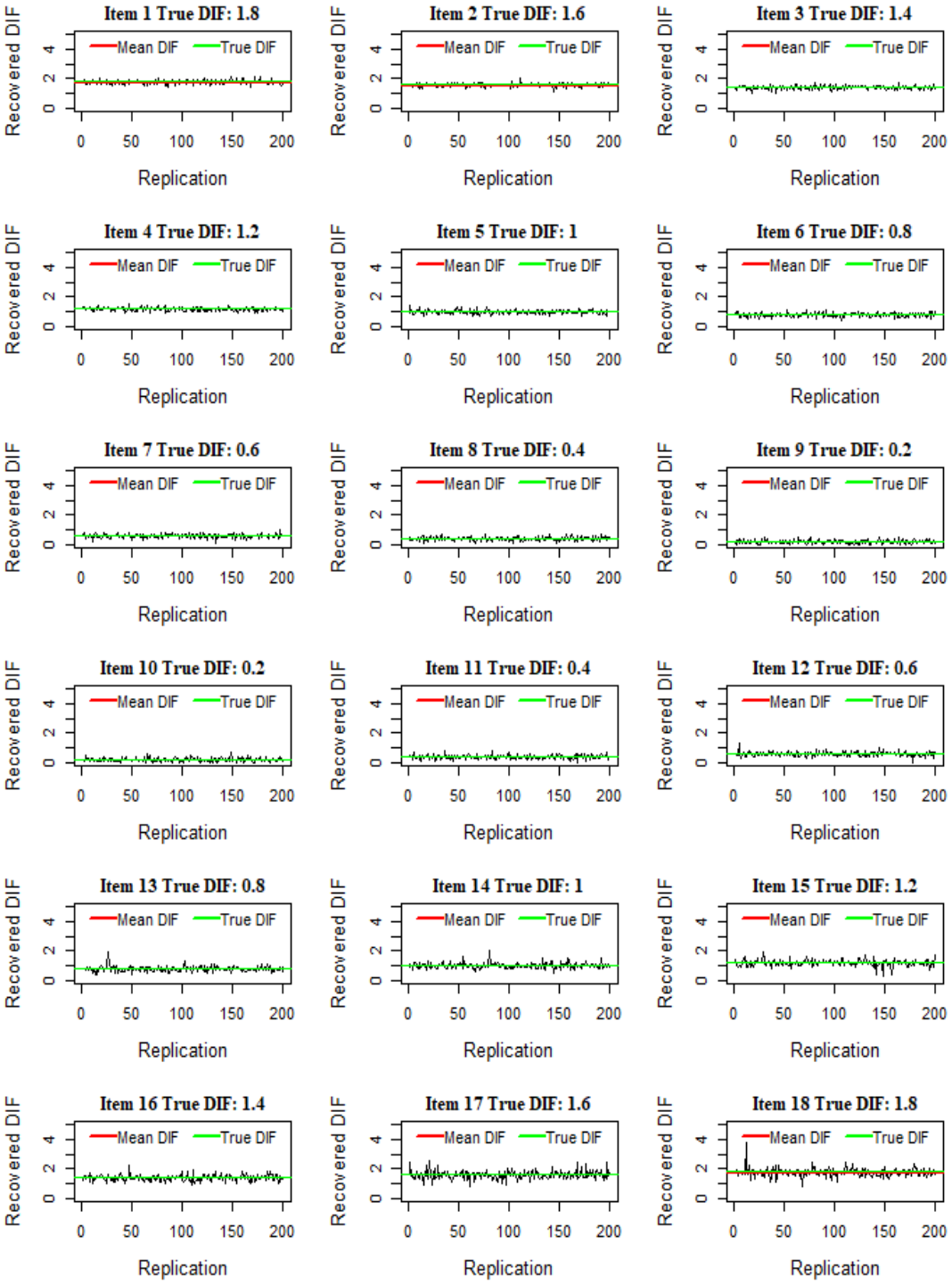

Figure 13
1002g_lc2_e Item DIF Recovery

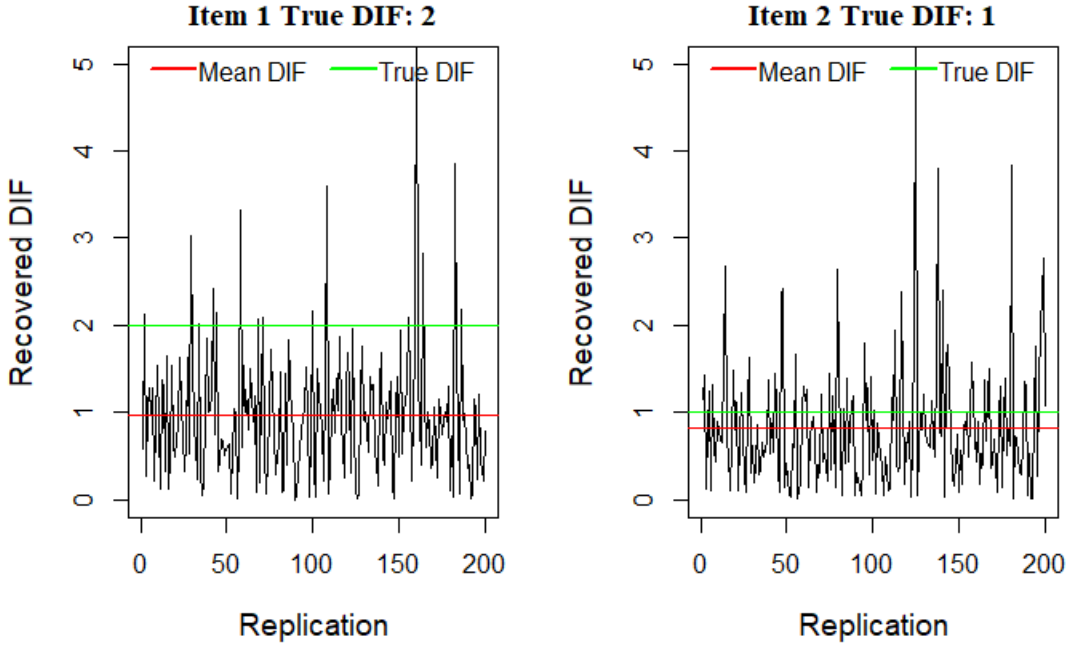

Figure 14
1004g_lc2_e Item DIF Recovery

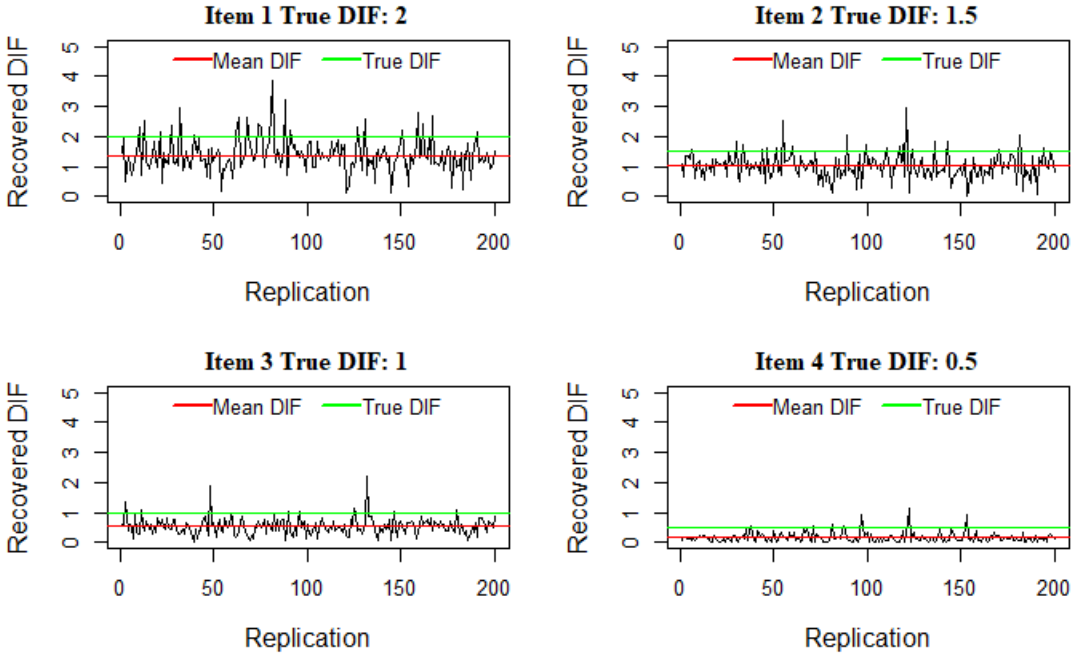

Figure 15
1006g_lc2_e Item DIF Recovery

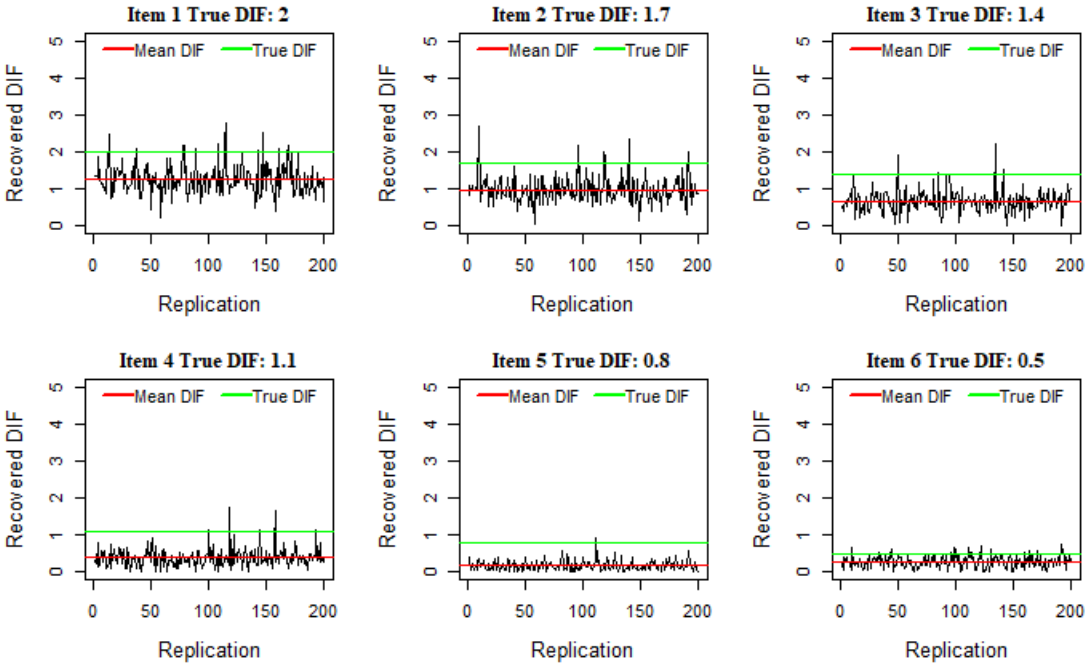

Figure 16
3006g_lc2_e Item DIF Recovery

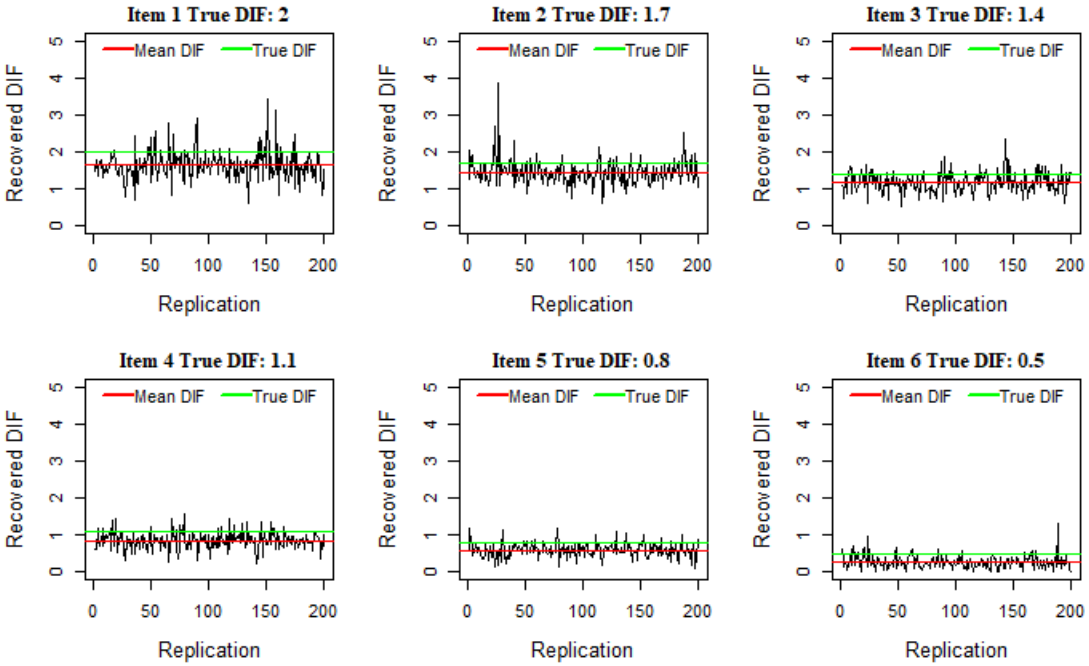

Figure 17
3012g_lc2_e Item DIF Recovery

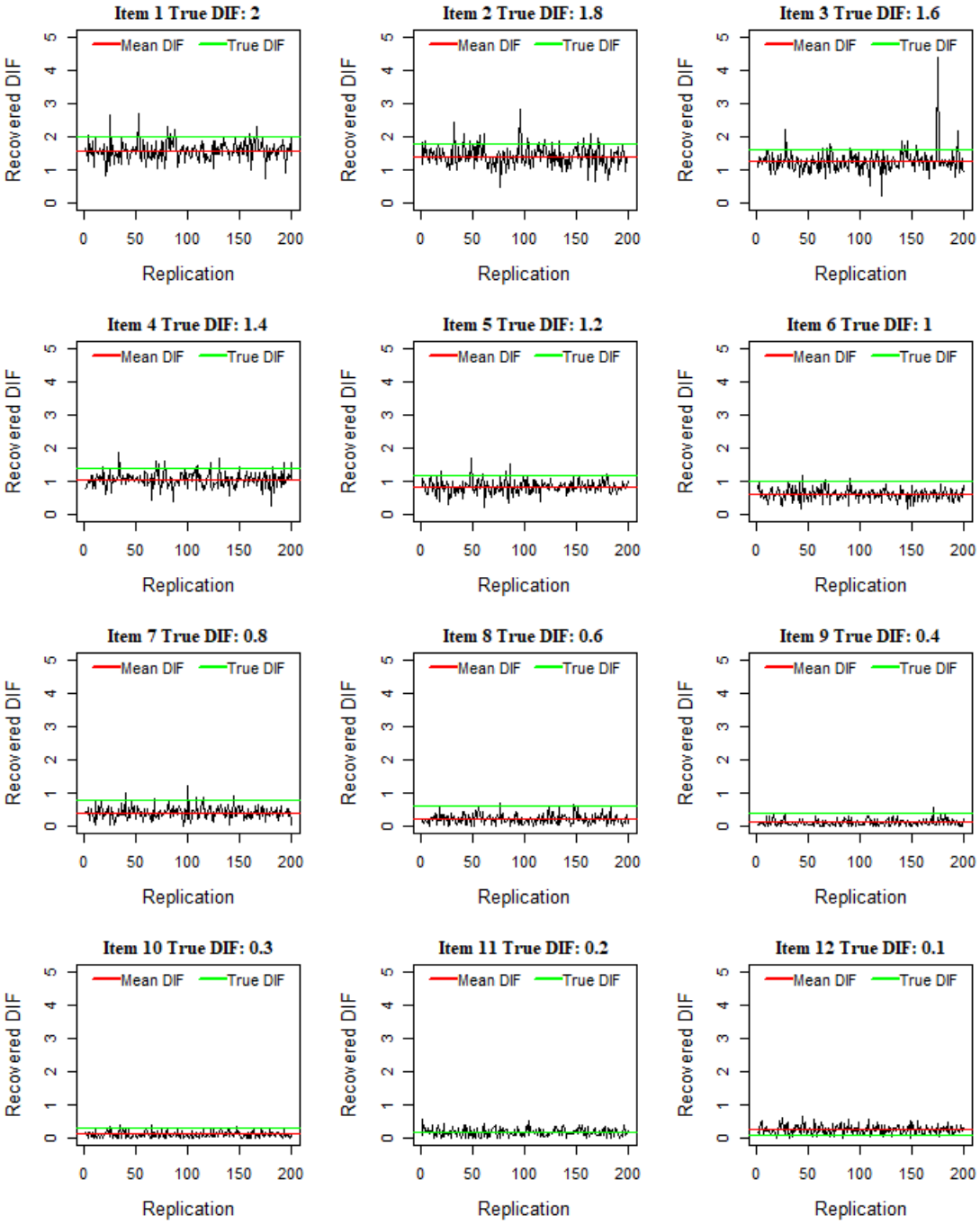

Figure 18

3018g_lc2_e Item DIF Recovery

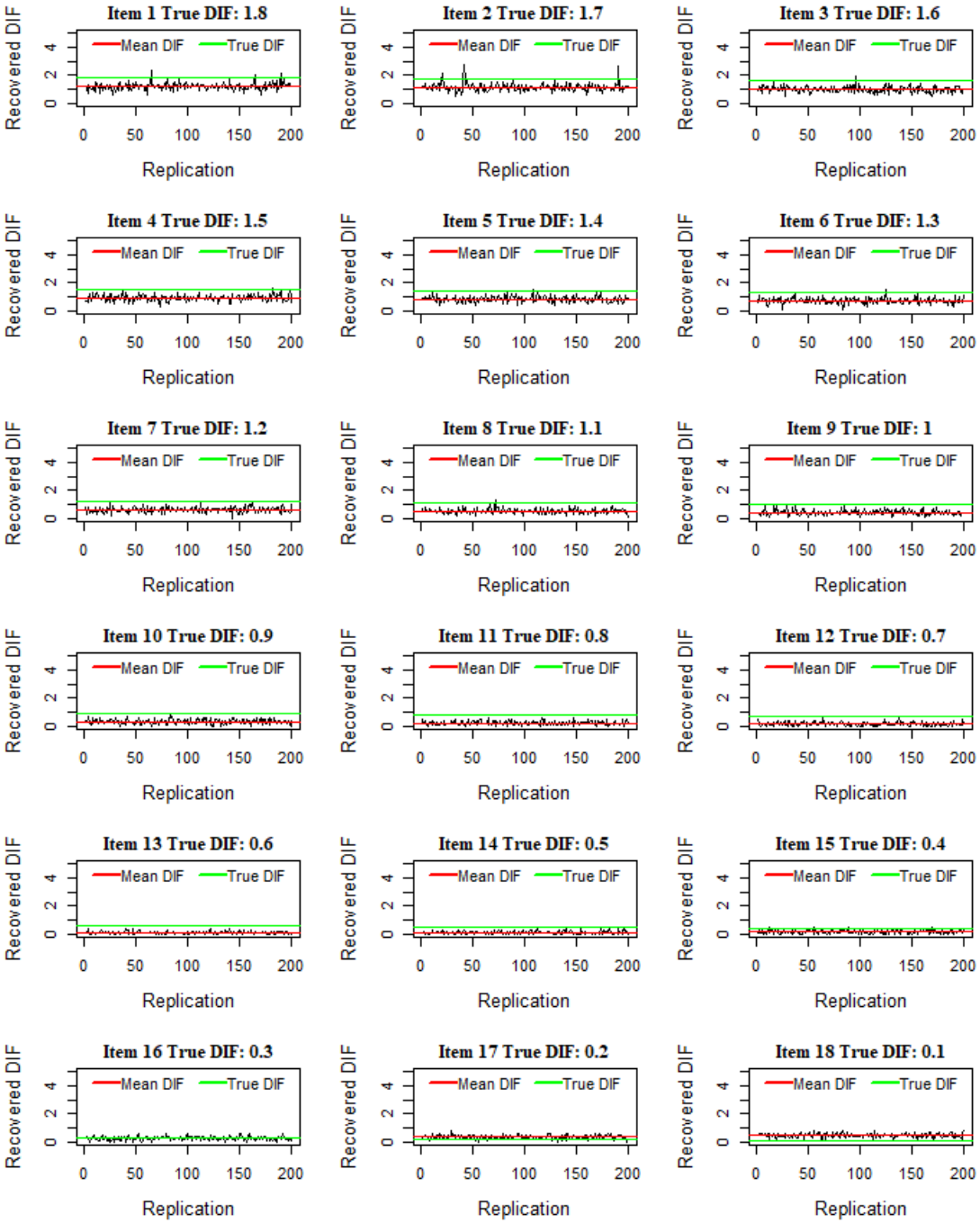

Figure 19
1002g_lc2_u Item DIF Recovery

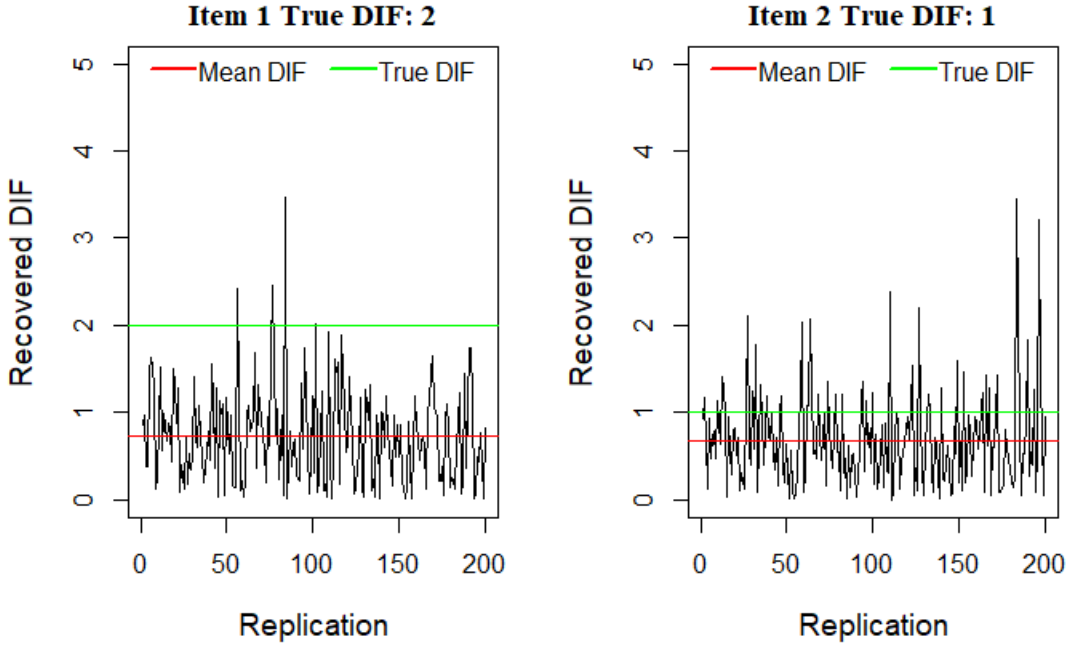

Figure 20
1004g_lc2_u Item DIF Recovery

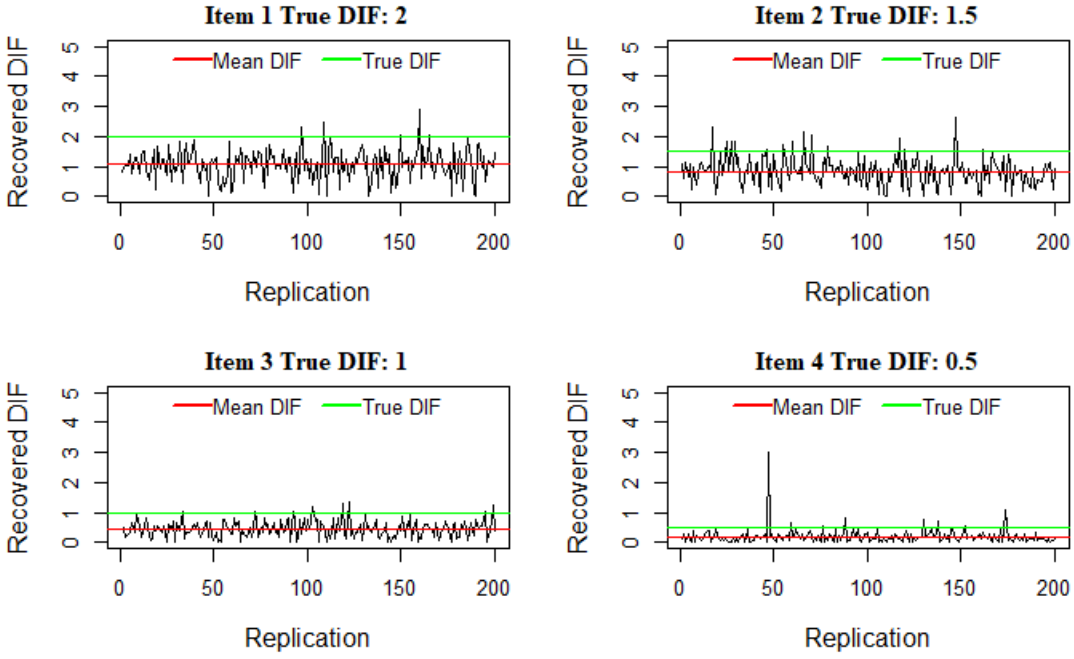

Figure 21
1006g_lc2_u Item DIF Recovery

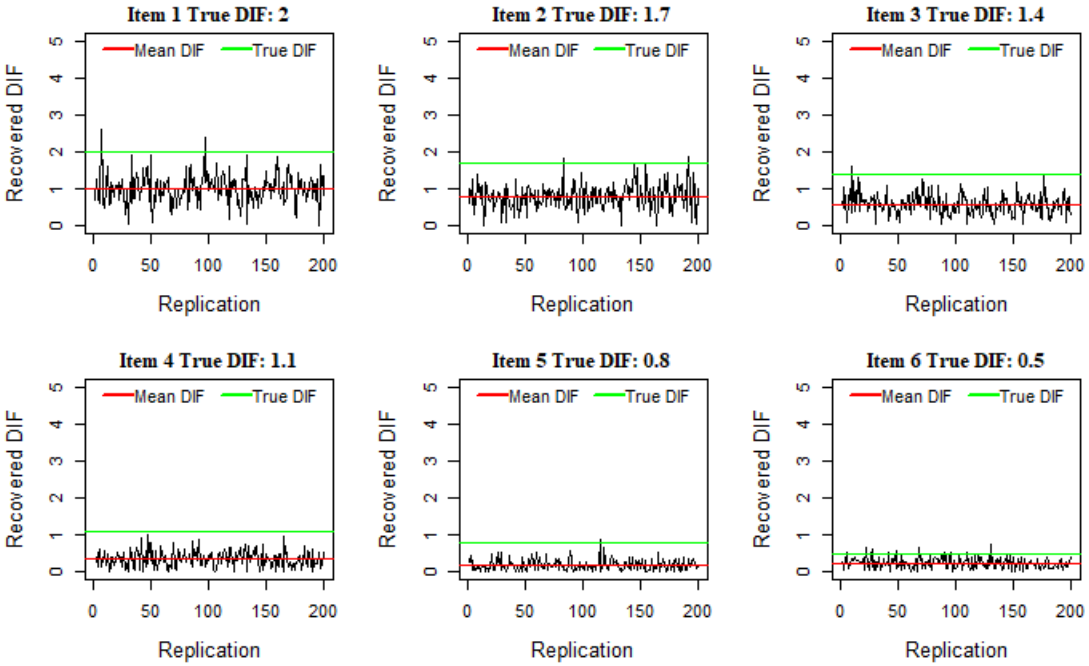

Figure 22
3006g_lc2_u Item DIF Recovery

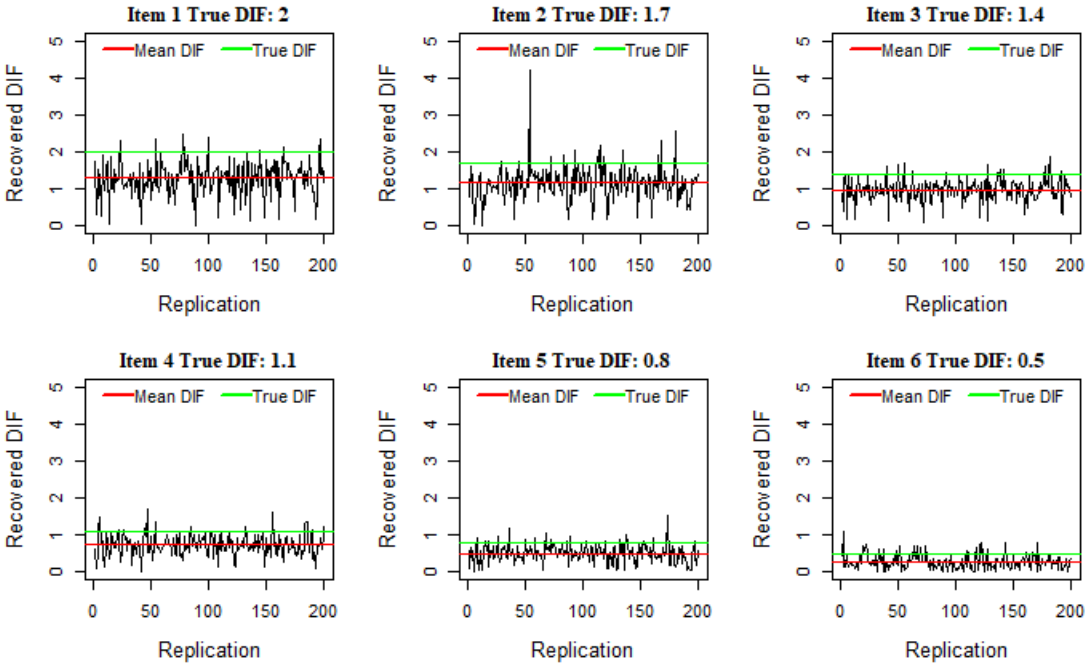

Figure 23

3012g_lc2_u Item DIF Recovery

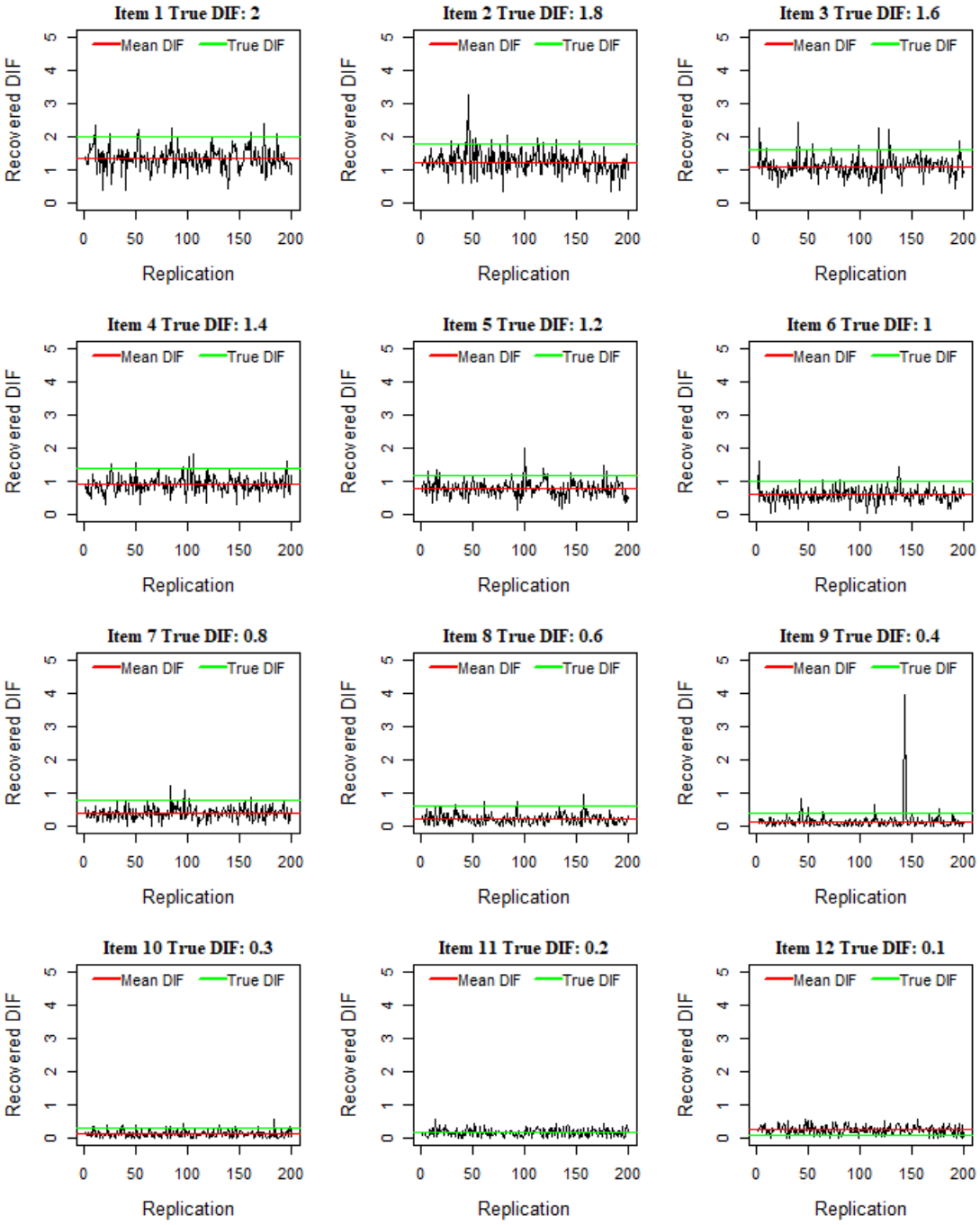

Figure 24
3018g_lc2_u Item DIF Recovery

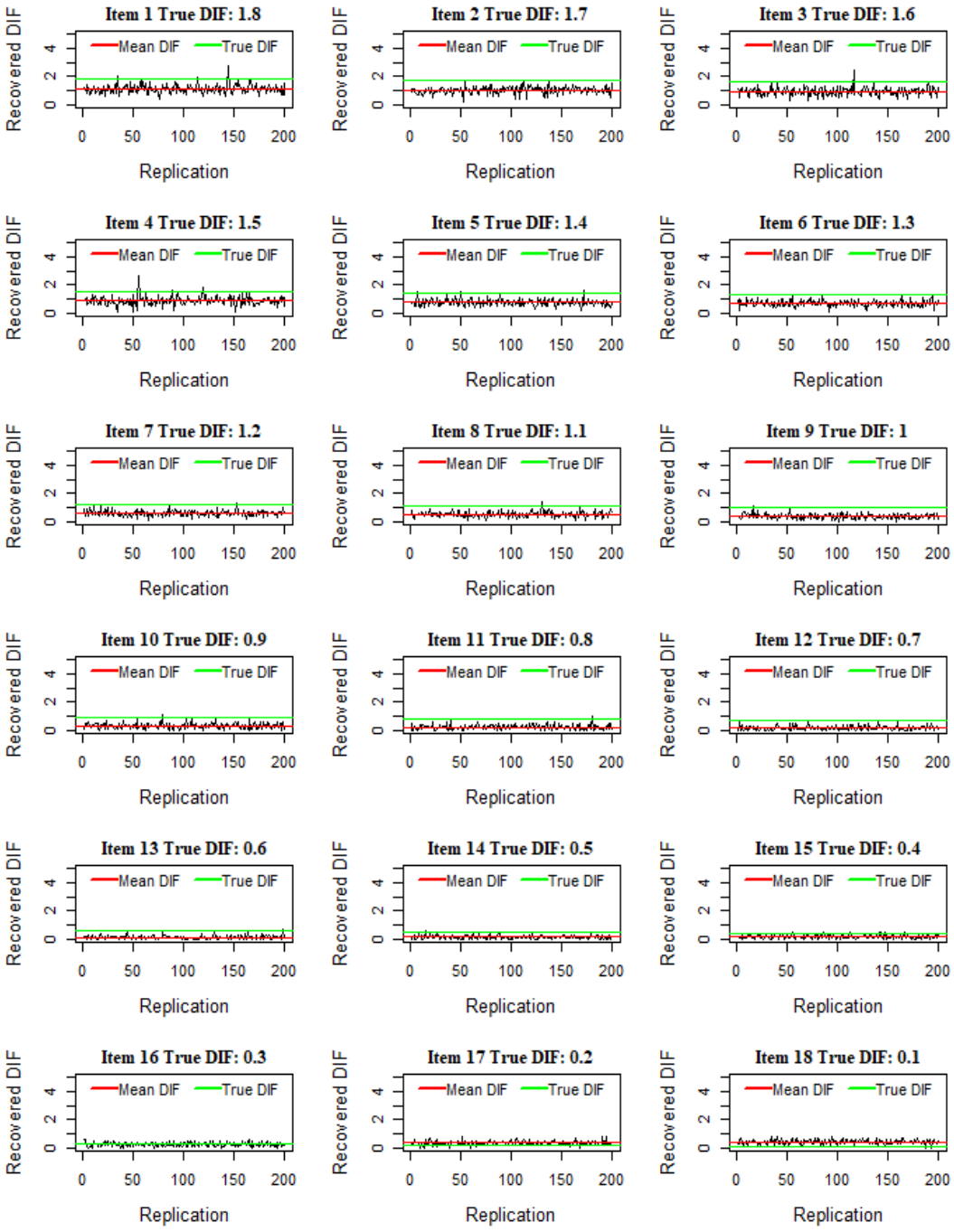

Figure 25a
1002s_lc3_e Item DIF Recovery

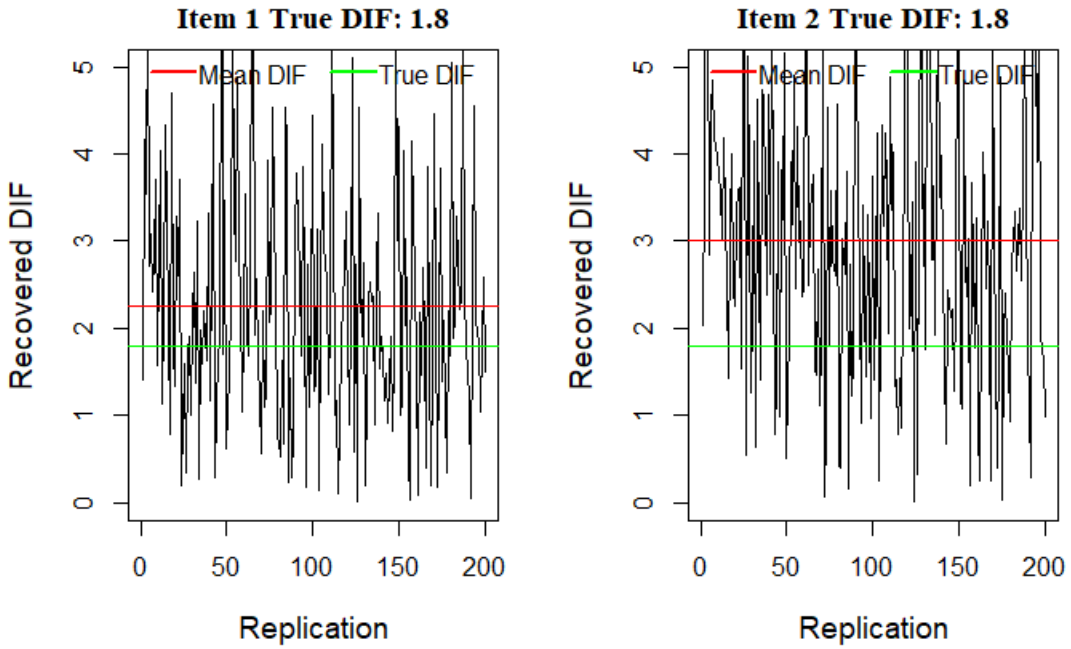

Figure 25b
1002s_lc3_e Item DIF Recovery

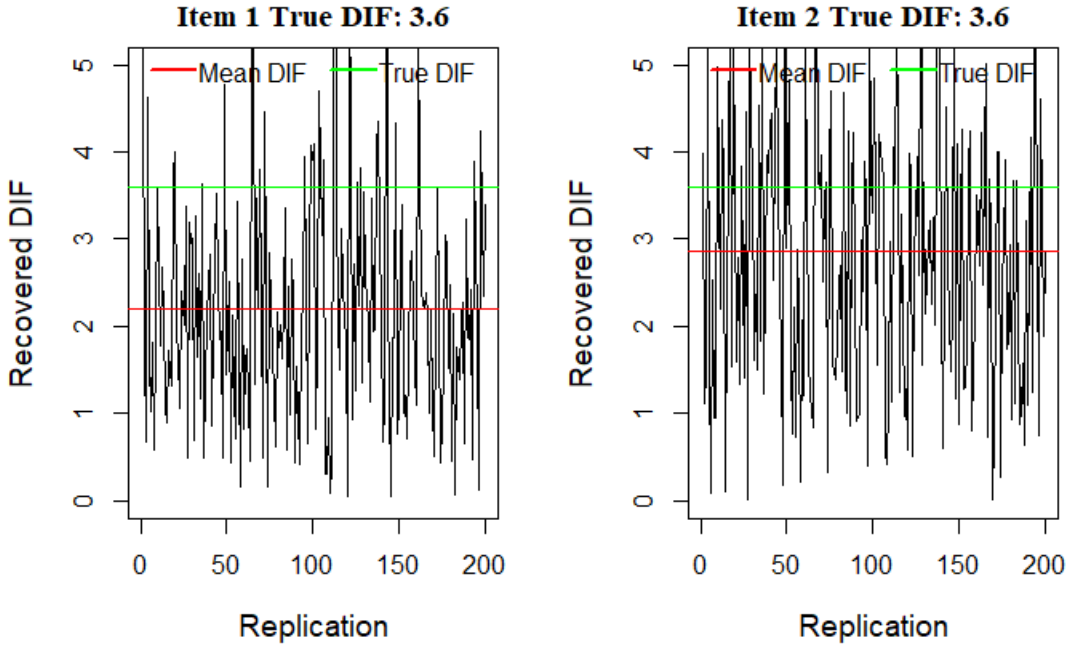

Figure 26a
1004s_lc3_e Item DIF Recovery

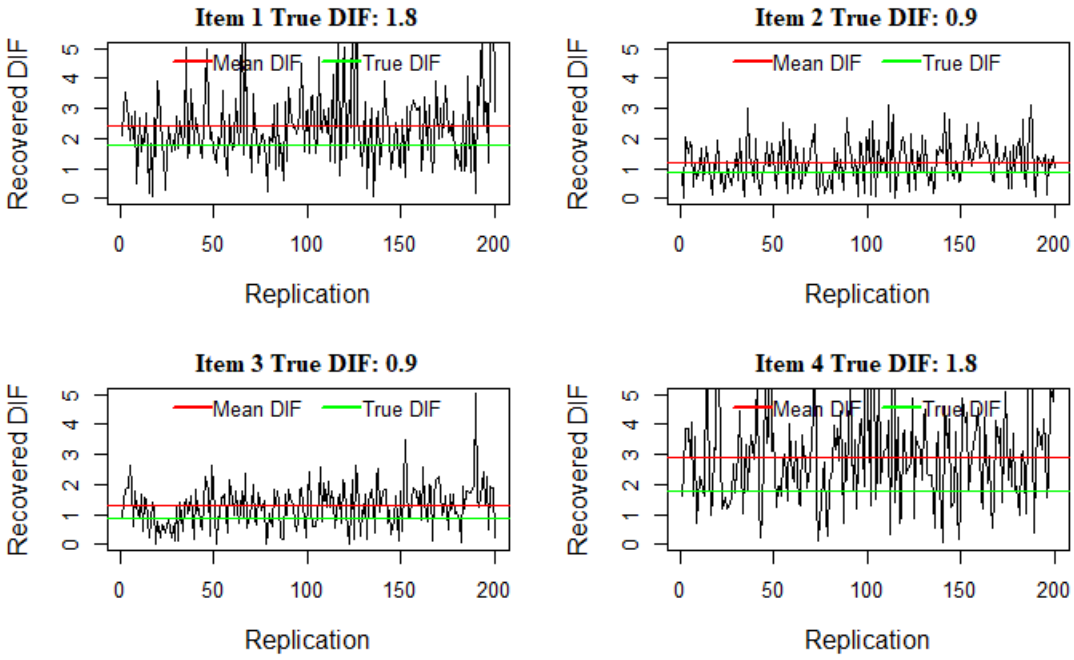

Figure 26b
1004s_lc3_e Item DIF Recovery

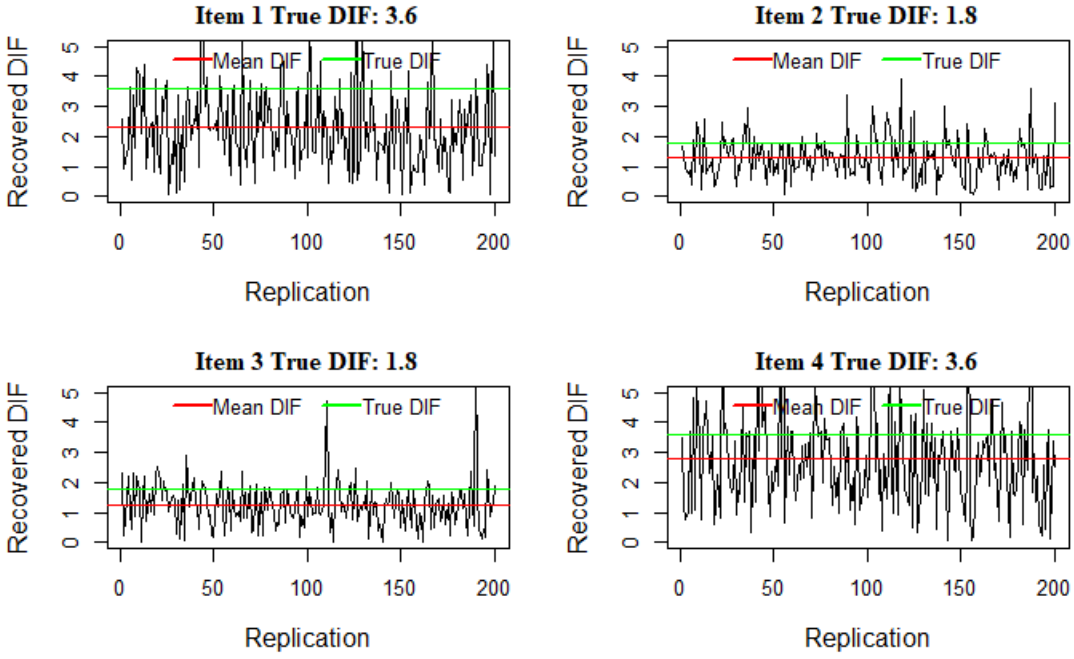

Figure 27a

1006s_lc3_e Item DIF Recovery

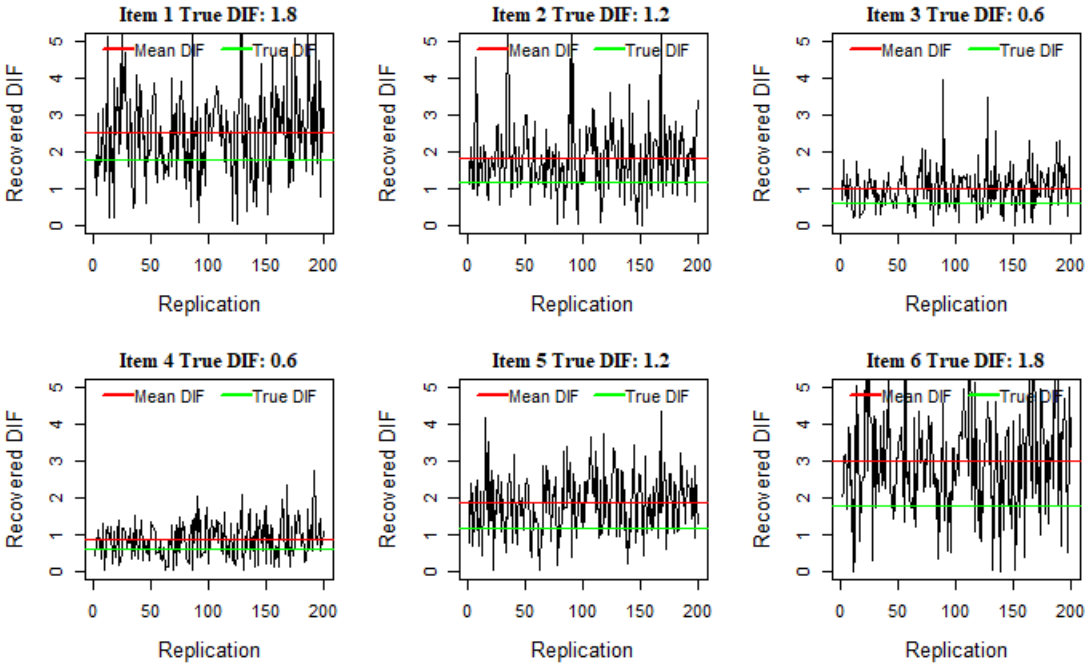

Figure 27b

1006s_lc3_e Item DIF Recovery

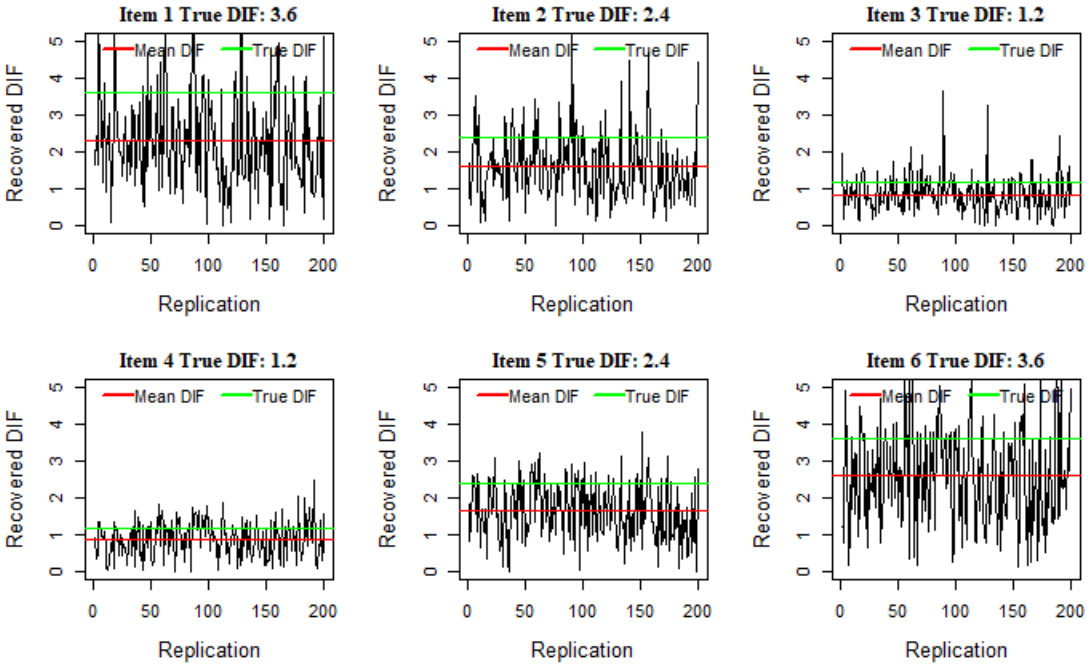

Figure 28a
3006s_lc3_e Item DIF Recovery

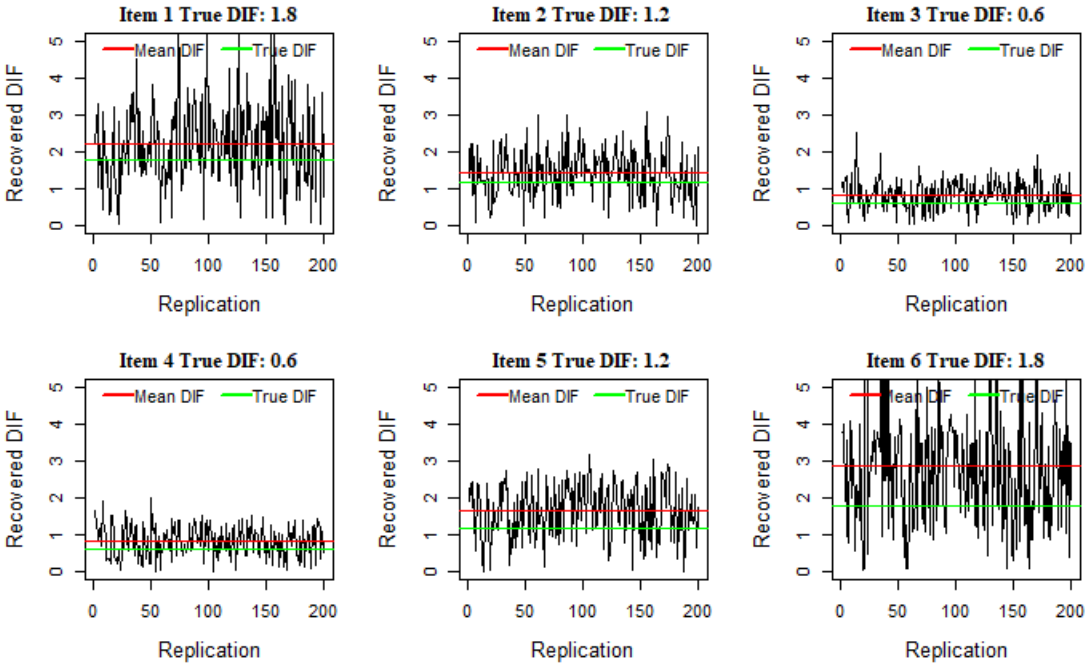

Figure 28b
3006s_lc3_e Item DIF Recovery

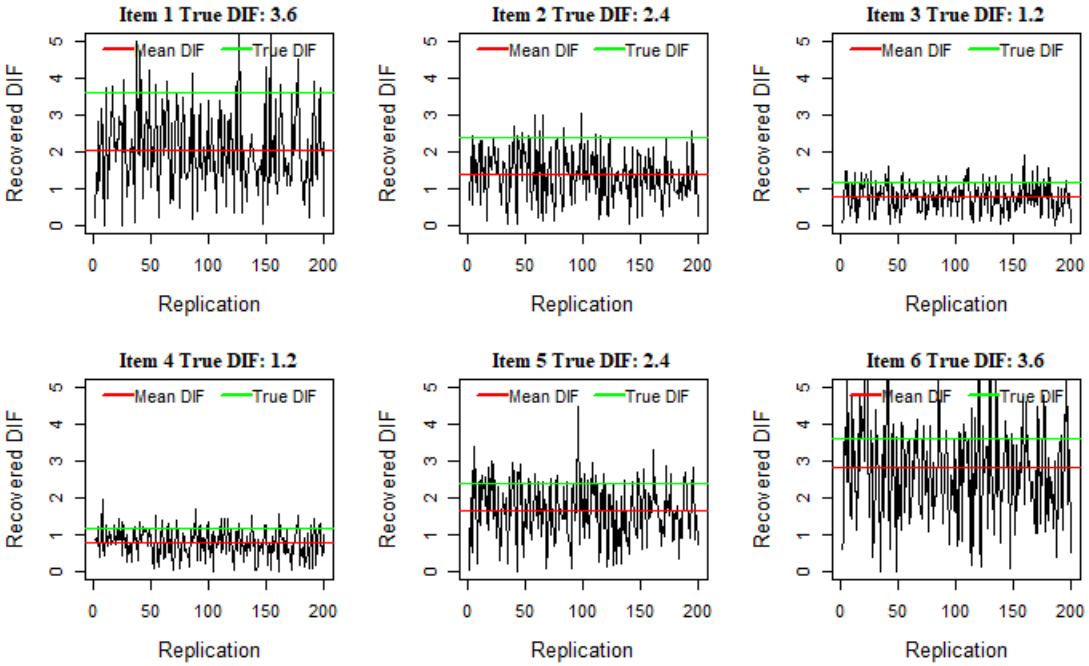

Figure 29a

3012s_lc3_e Item DIF Recovery

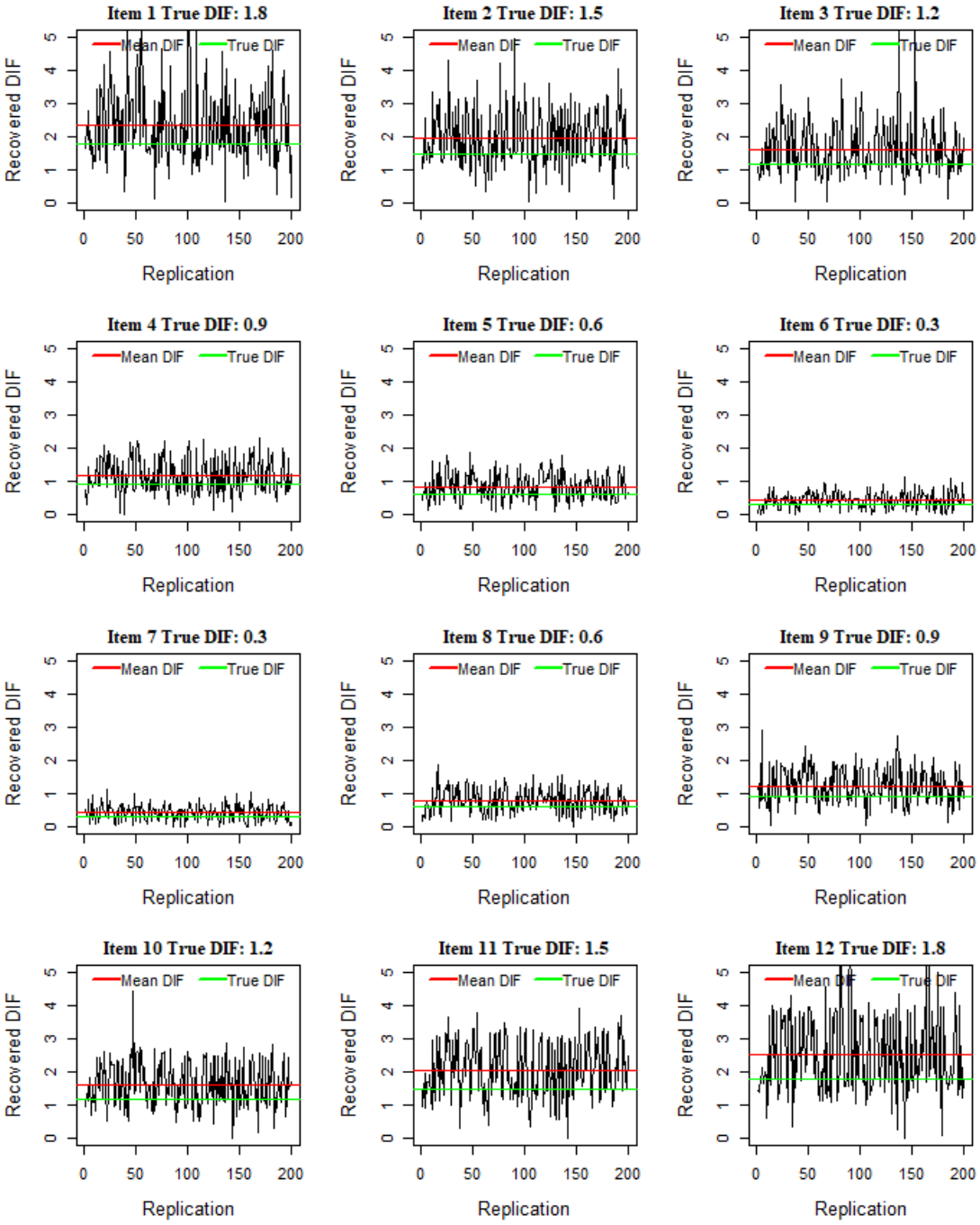

Figure 29b

3012s_lc3_e Item DIF Recovery

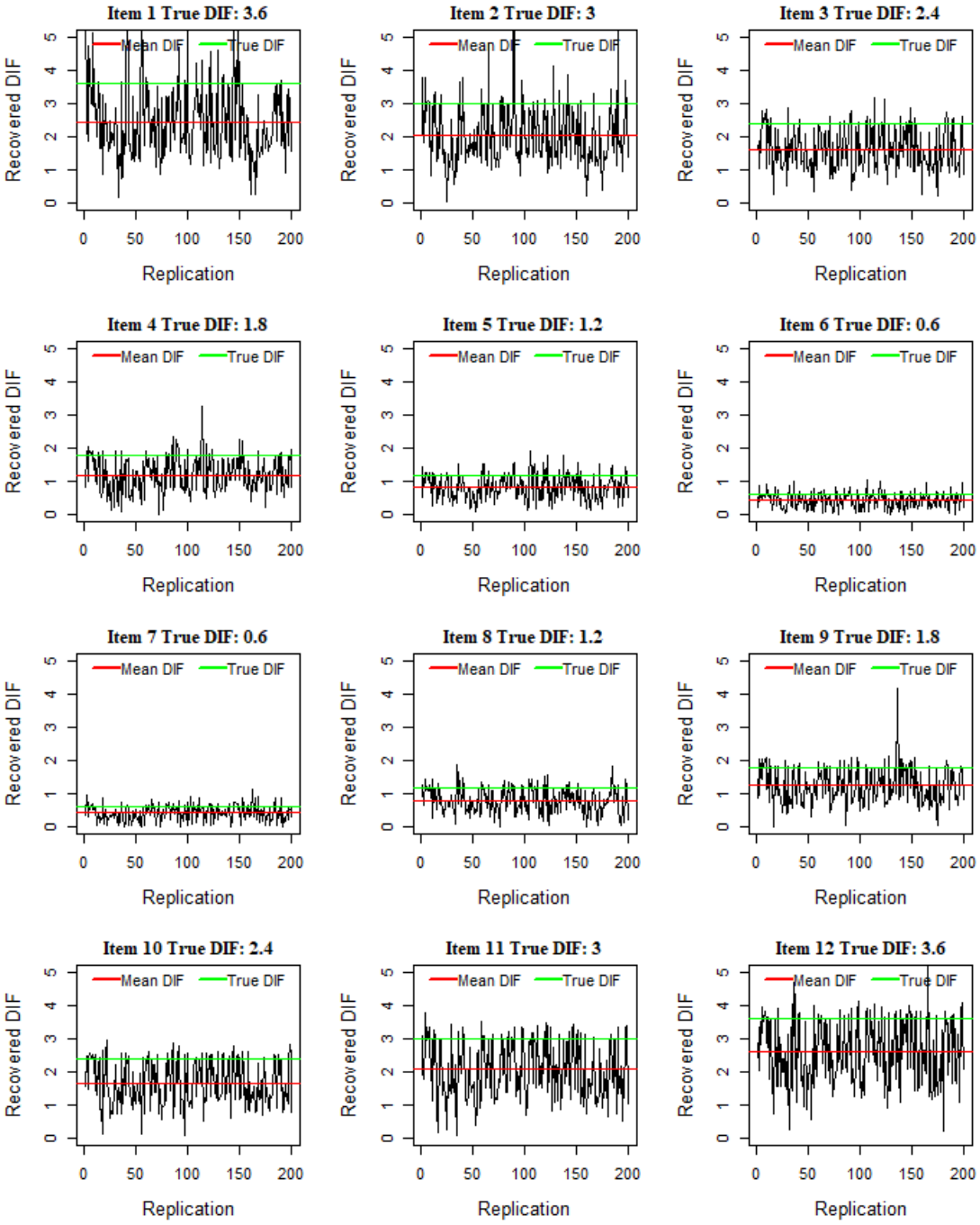

Figure 30a

3018s_lc3_e Item DIF Recovery

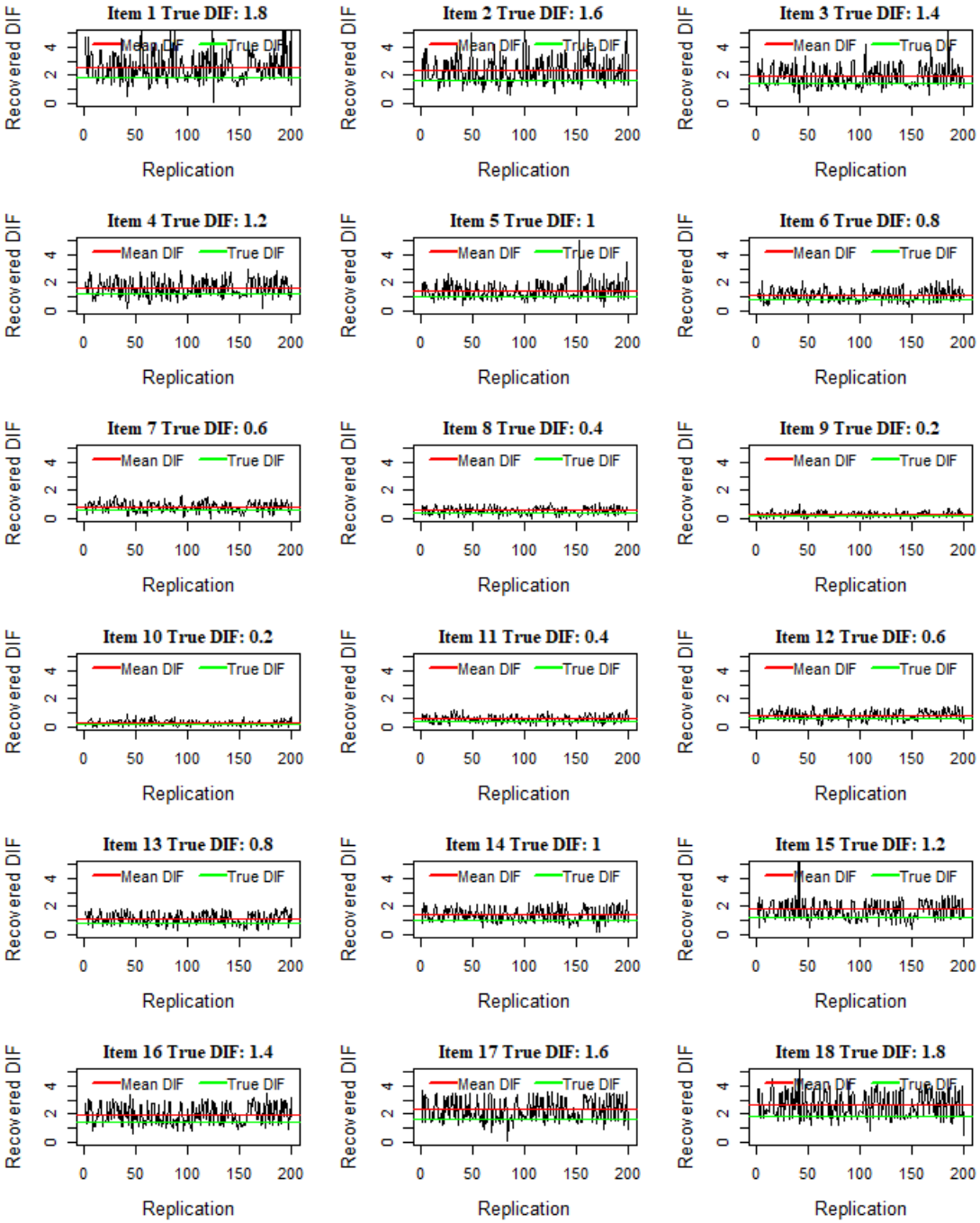

Figure 30b

3018s_lc3_e Item DIF Recovery

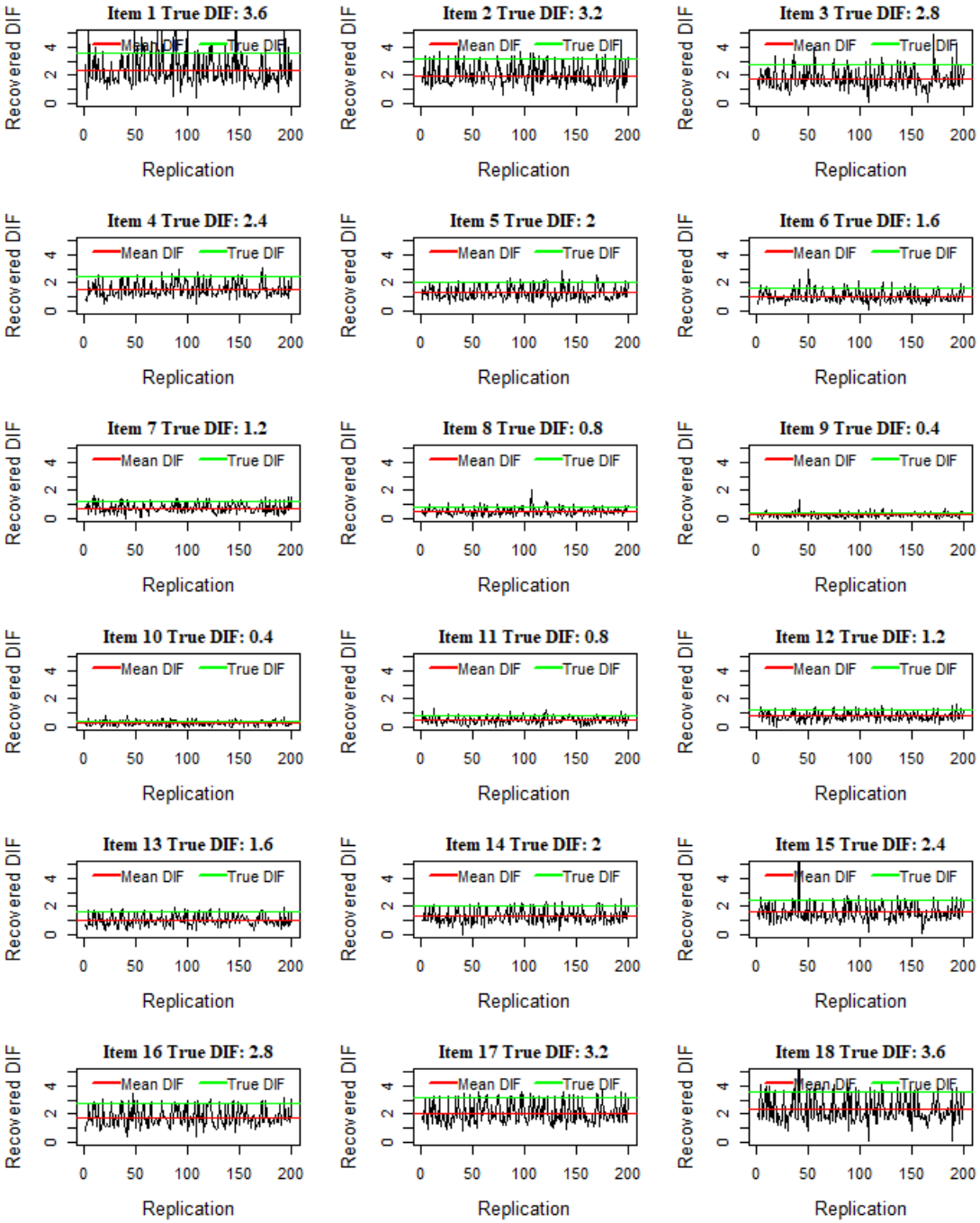

Figure 31a
1002s_lc3_u Item DIF Recovery

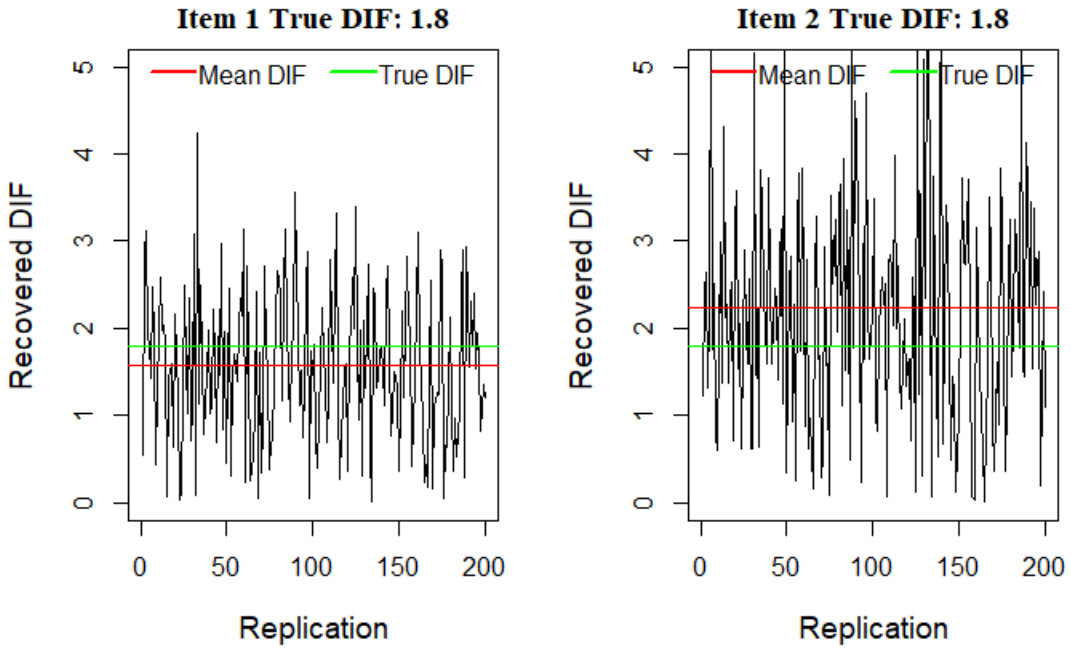

Figure 31b
1002s_lc3_u Item DIF Recovery

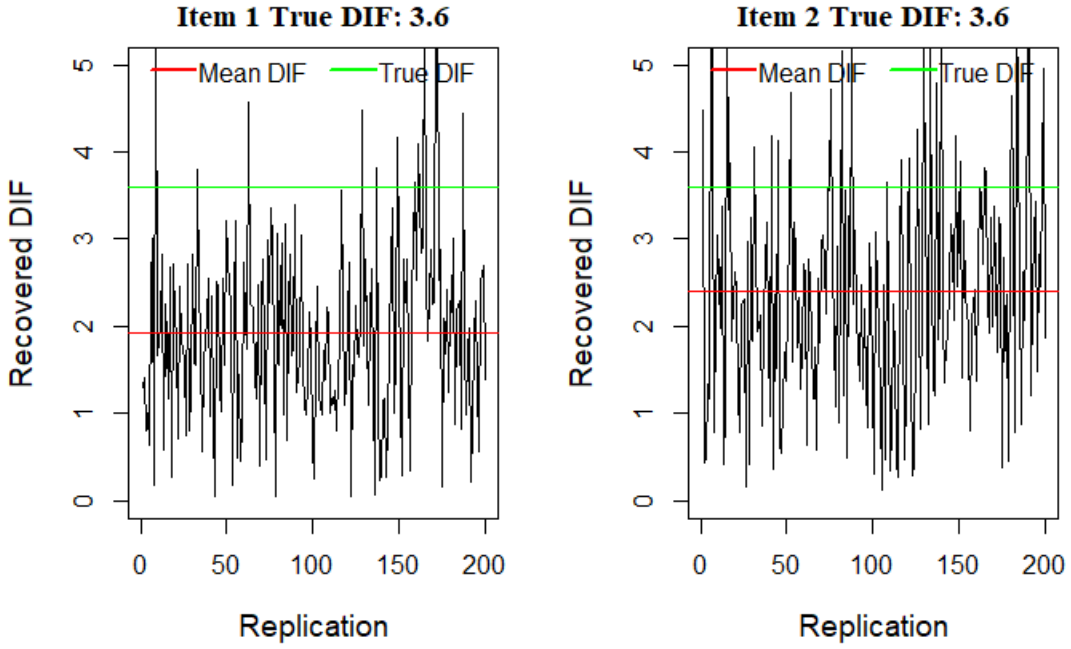

Figure 32a
1004s_lc3_u Item DIF Recovery

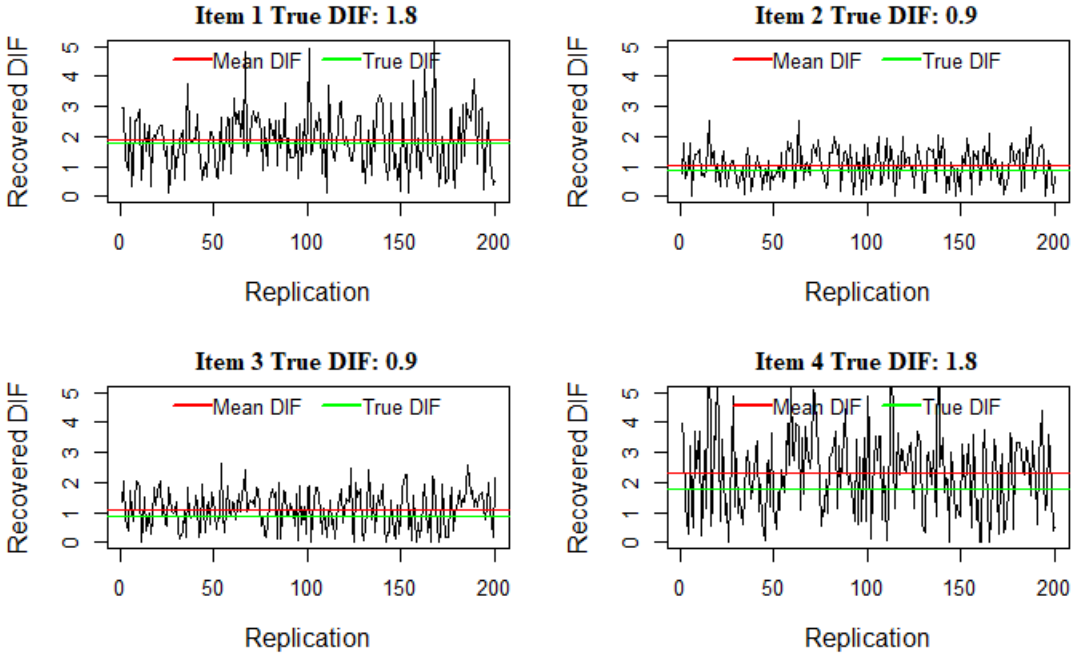

Figure 32b
1004s_lc3_u Item DIF Recovery

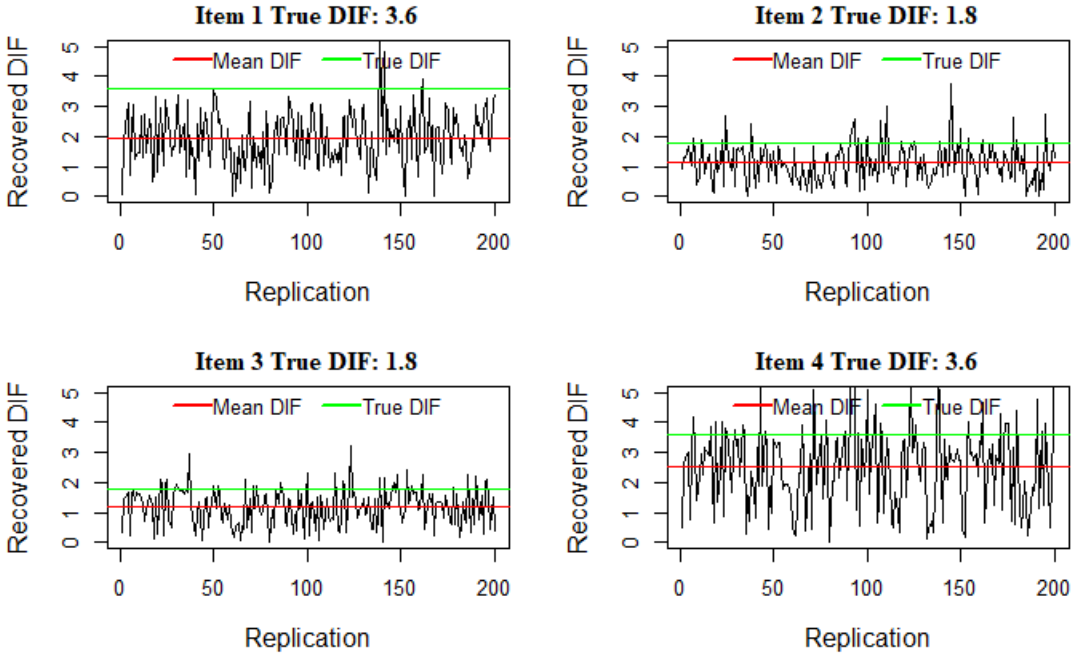

Figure 33a

1006s_lc3_u Item DIF Recovery

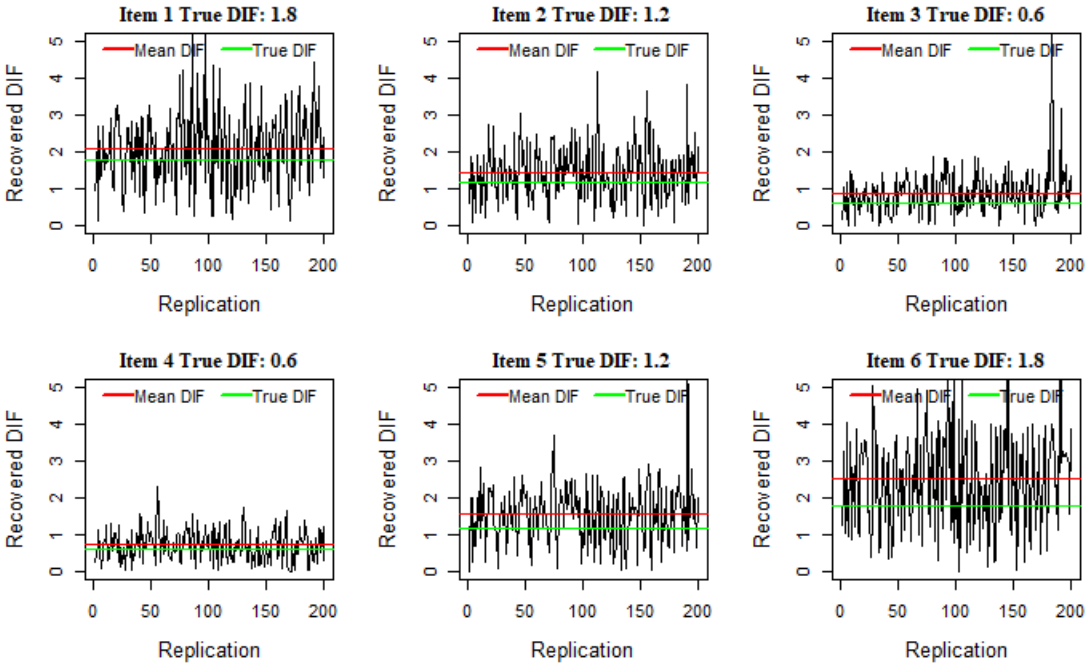

Figure 33b

1006s_lc3_u Item DIF Recovery

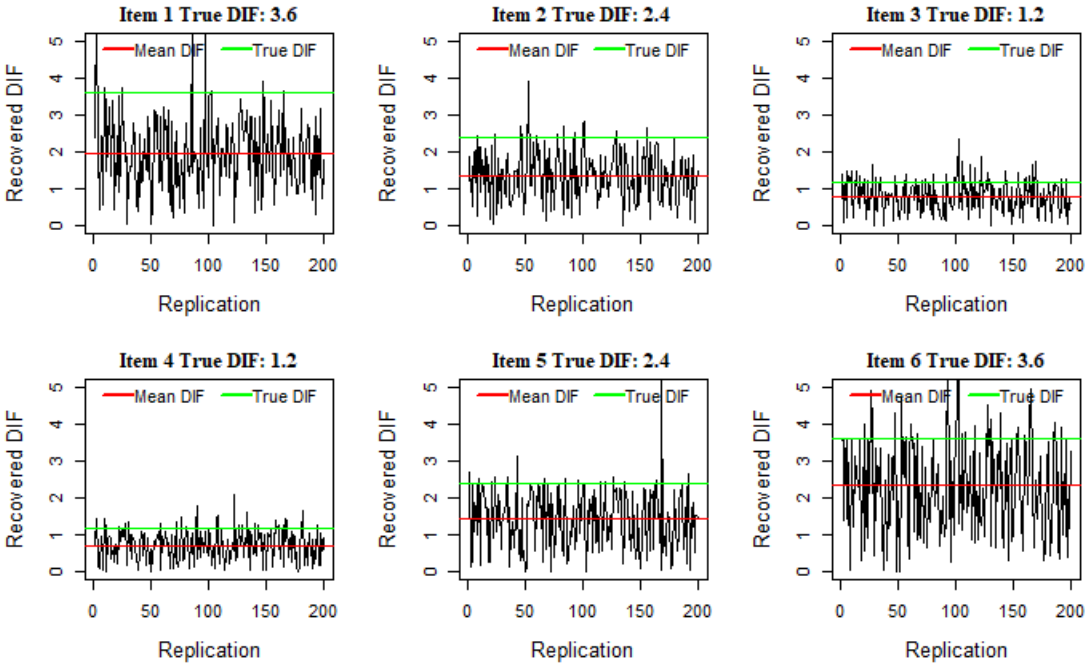

Figure 34a
3006s_lc3_u Item DIF Recovery

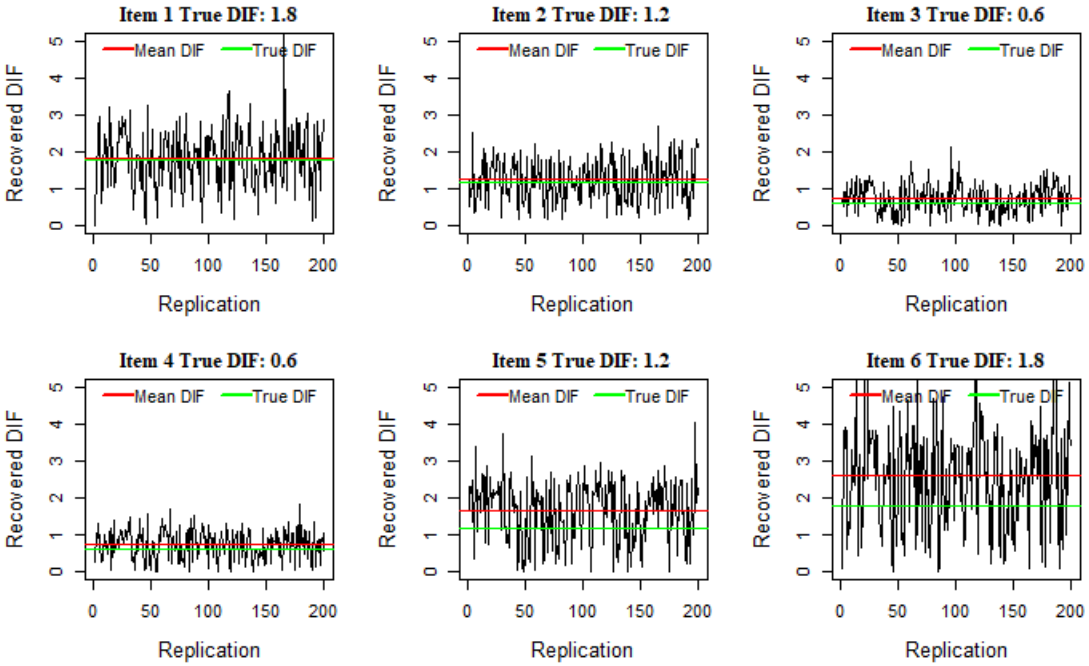

Figure 34b
3006s_lc3_u Item DIF Recovery

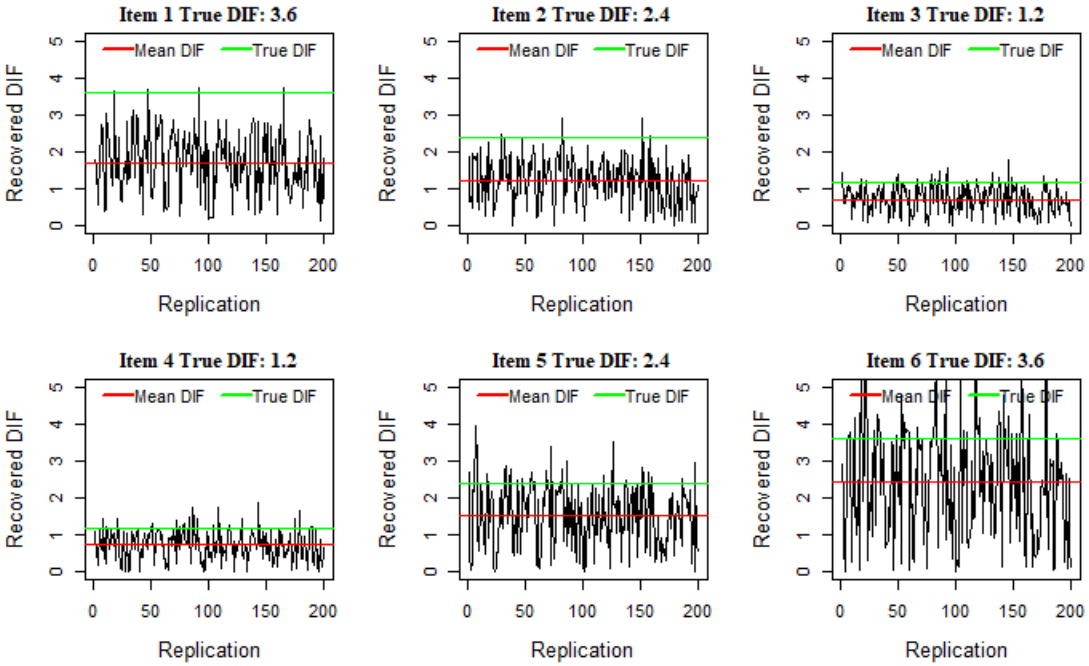

Figure 35a

3012s_lc3_u Item DIF Recovery

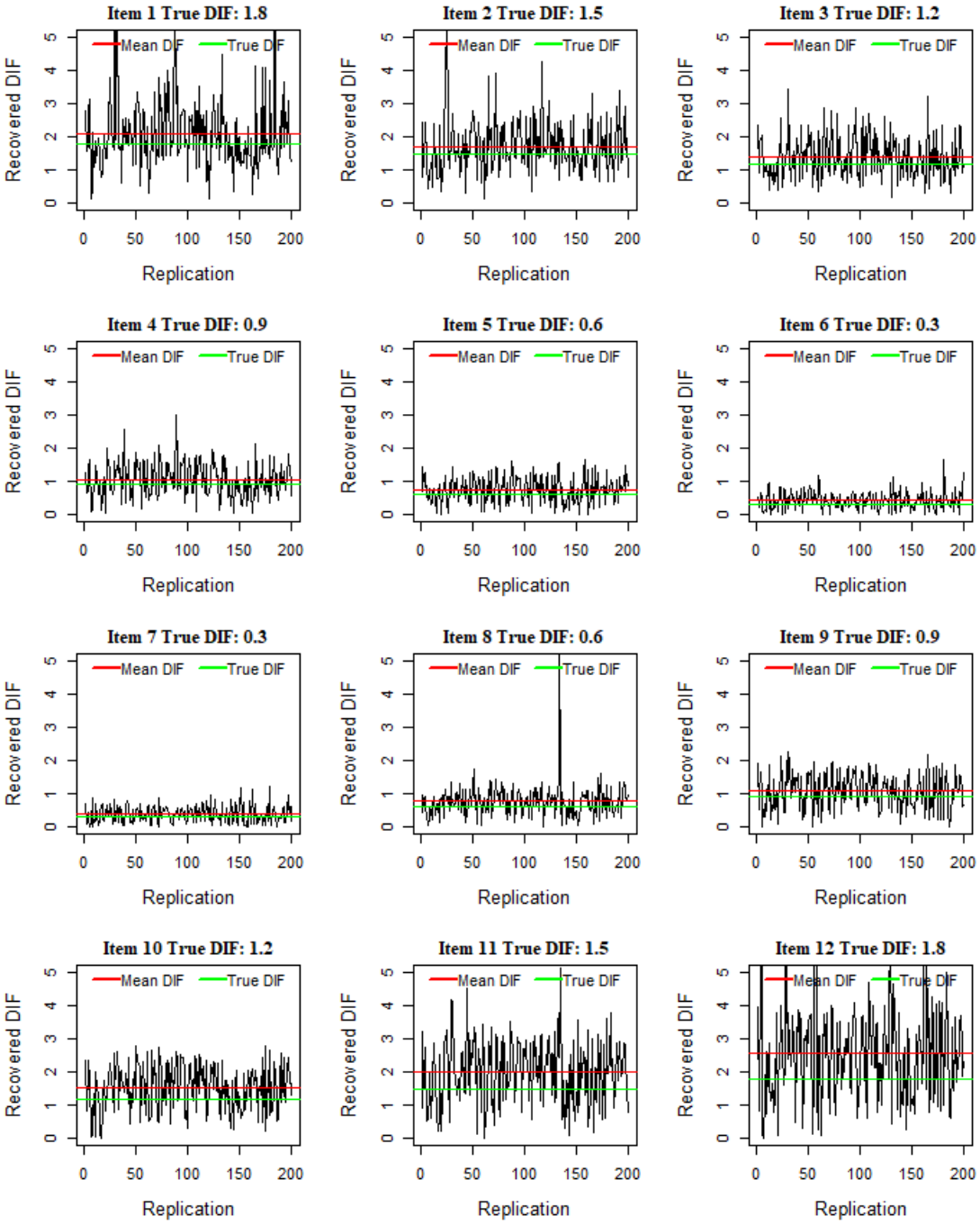

Figure 35b
3012s_lc3_u Item DIF Recovery

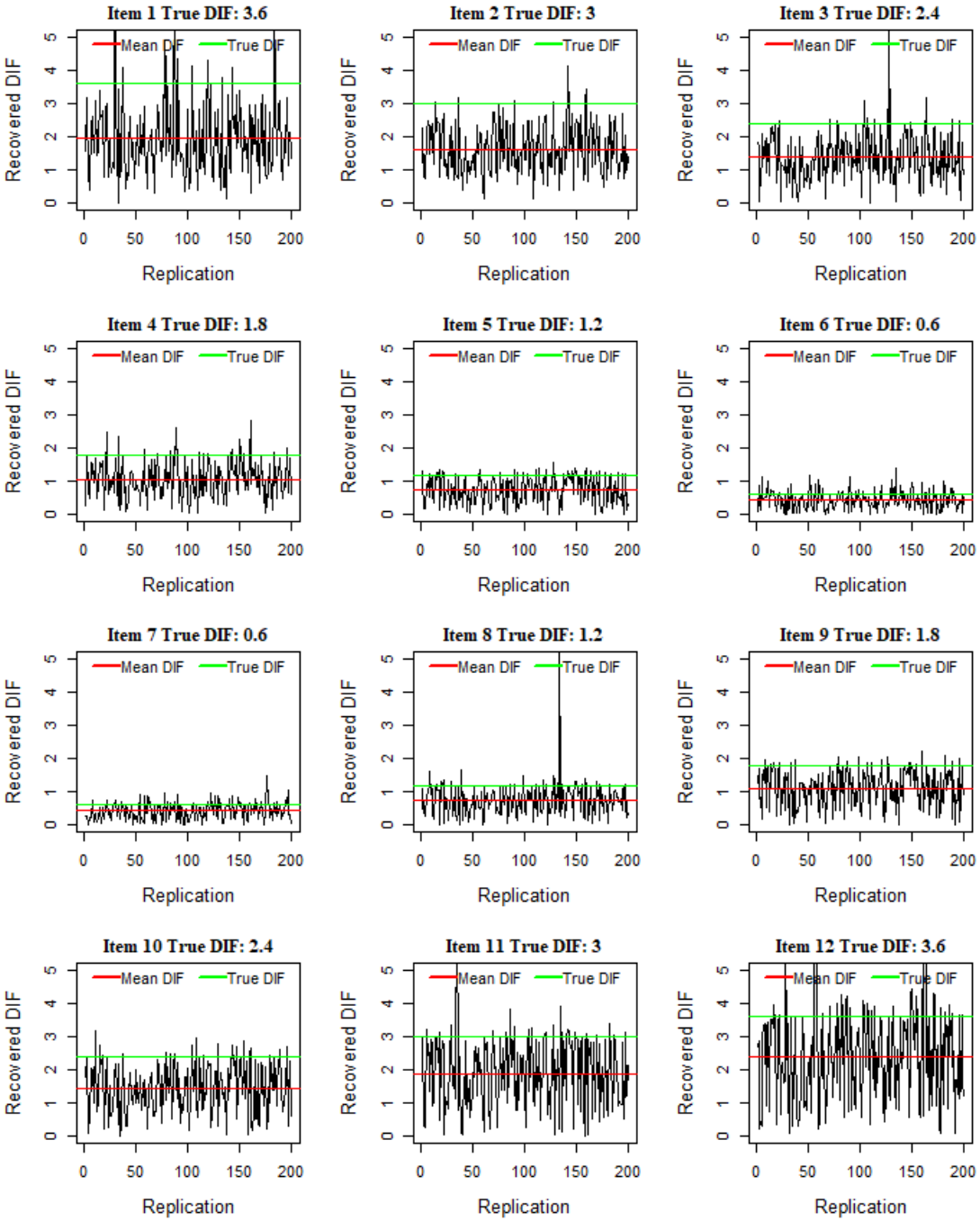

Figure 36a

3018s_lc3_u Item DIF Recovery

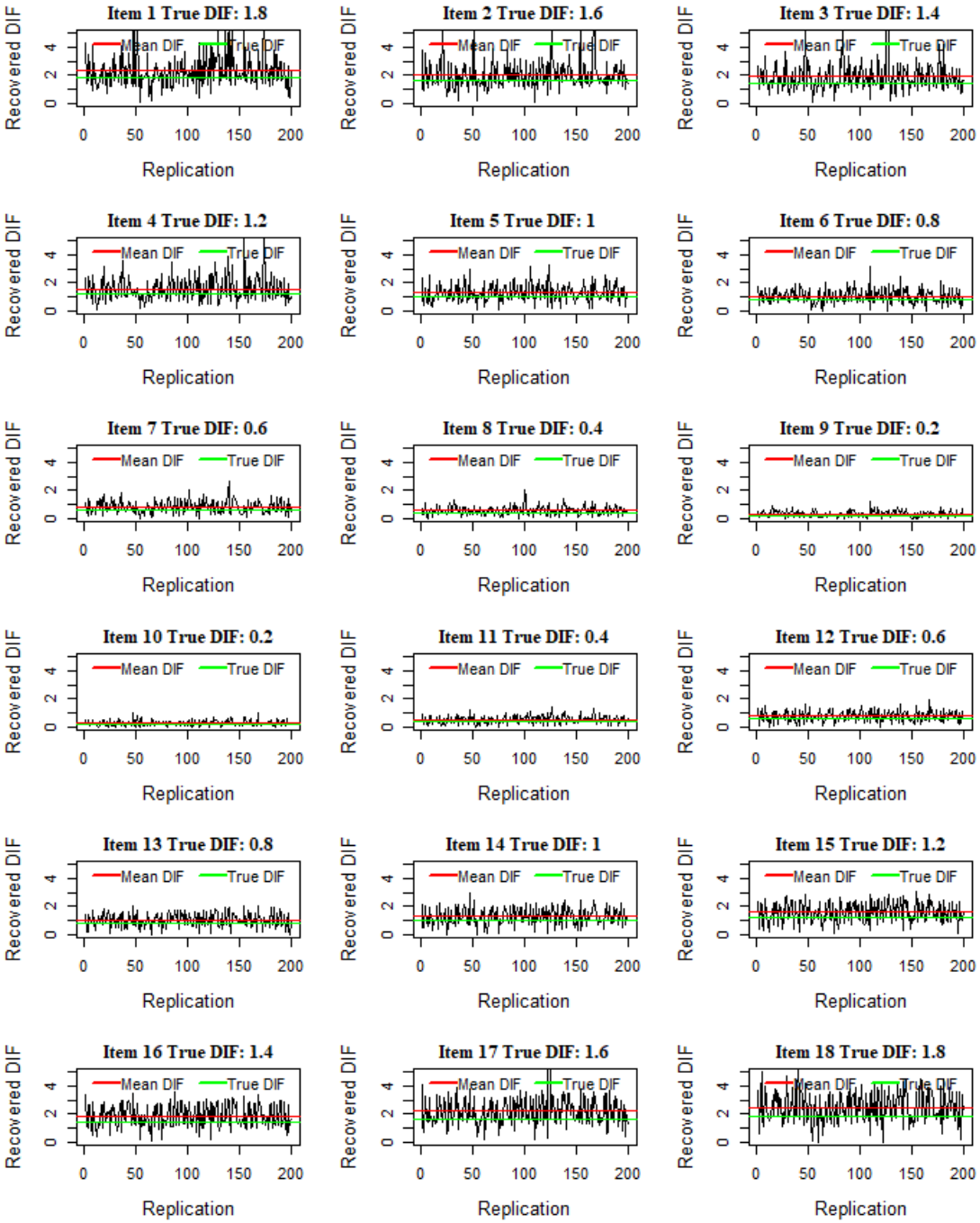

Figure 36b

3018s_lc3_u Item DIF Recovery

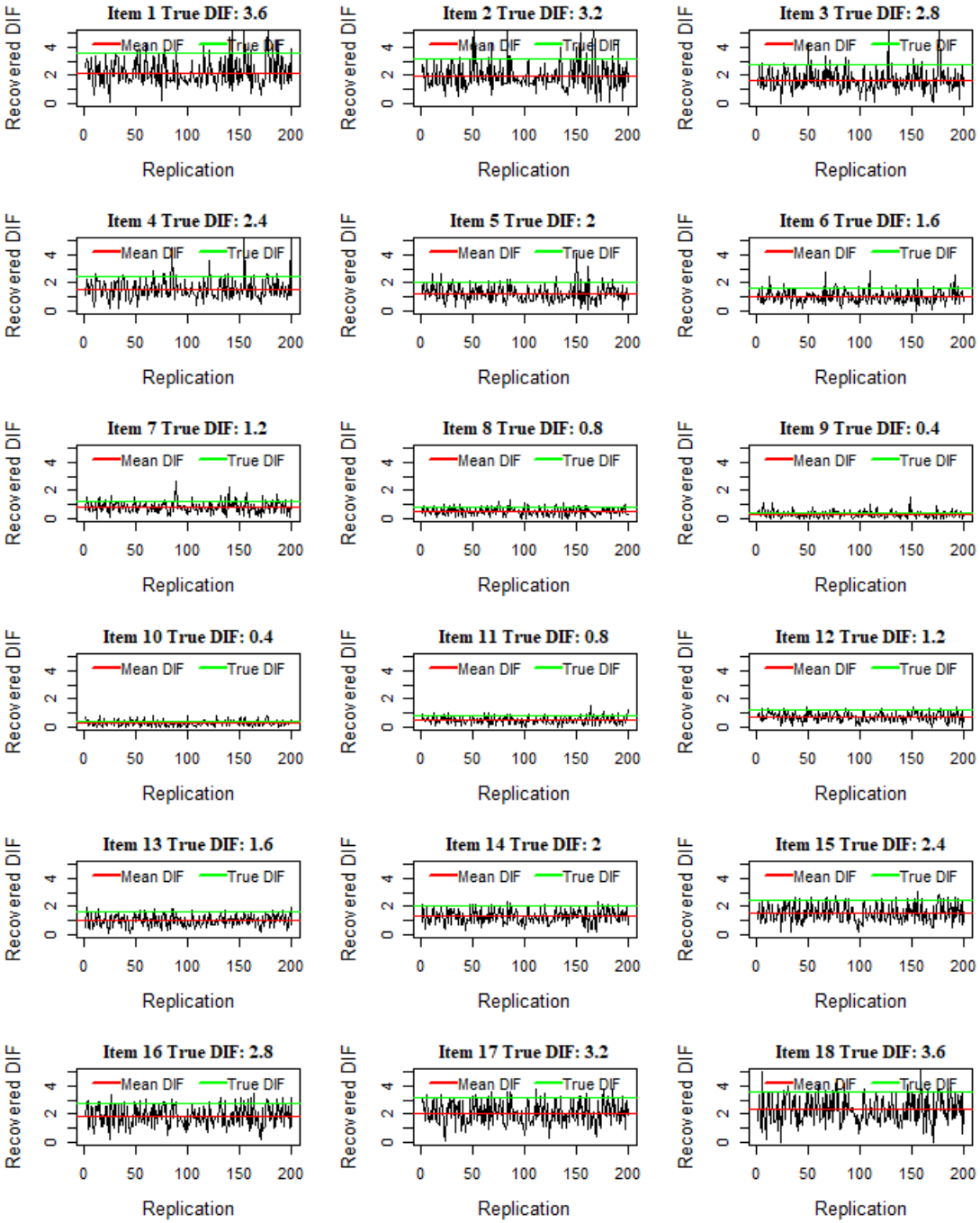

Figure 37a
1002g_lc3_e Item DIF Recovery

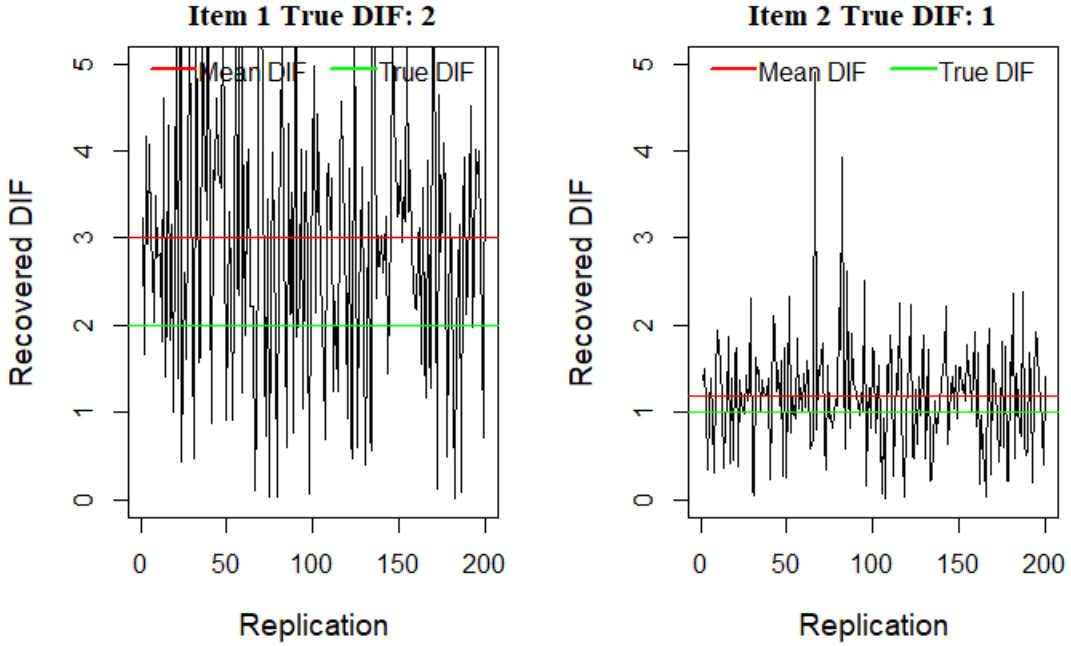

Figure 37b
1002g_lc3_e Item DIF Recovery

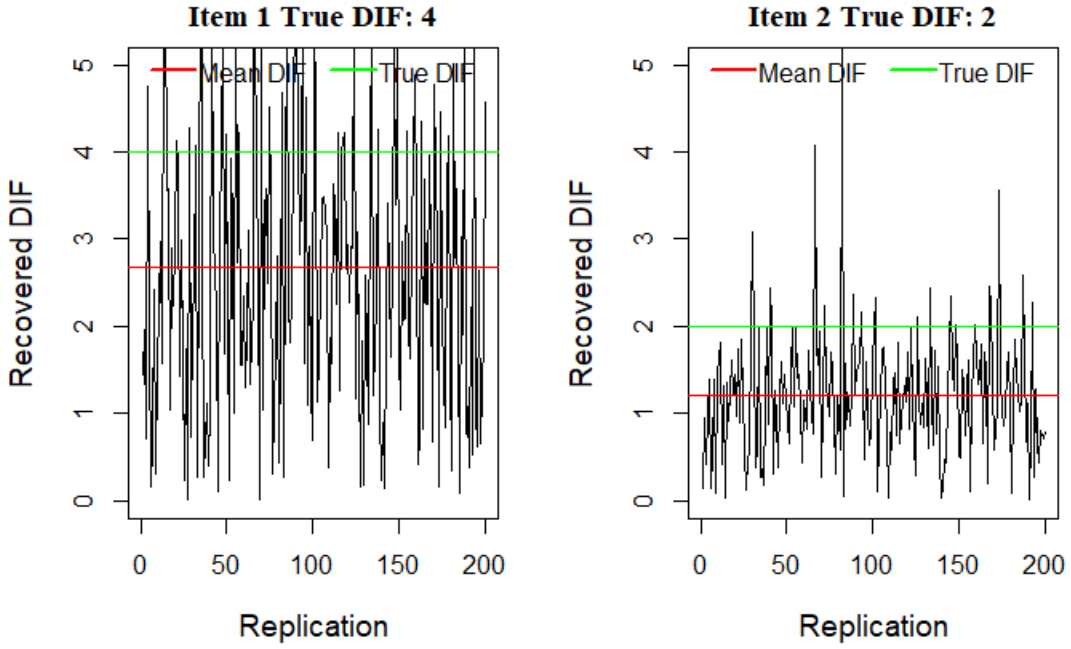

Figure 38a
1004g_lc3_e Item DIF Recovery

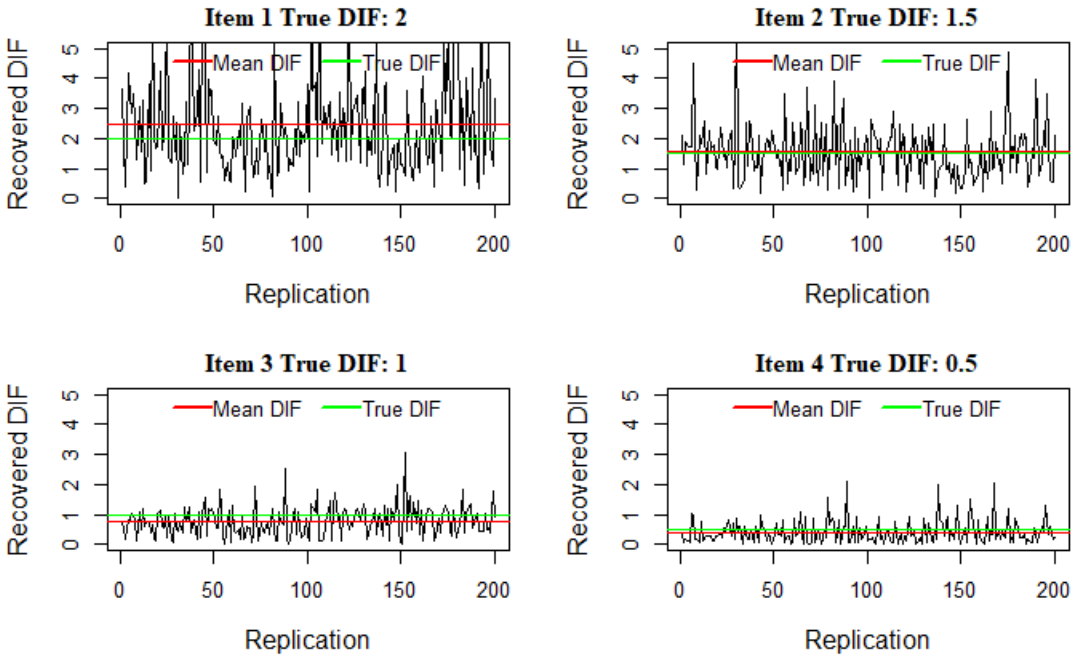

Figure 38b
1004g_lc3_e Item DIF Recovery

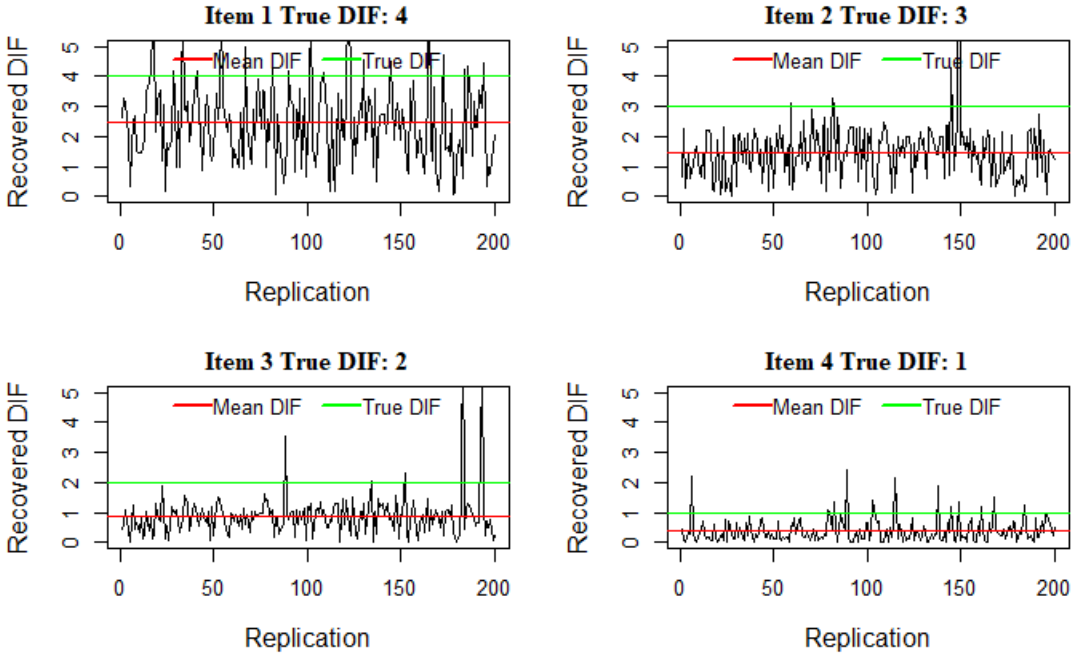

Figure 39a

1006g_lc3_e Item DIF Recovery

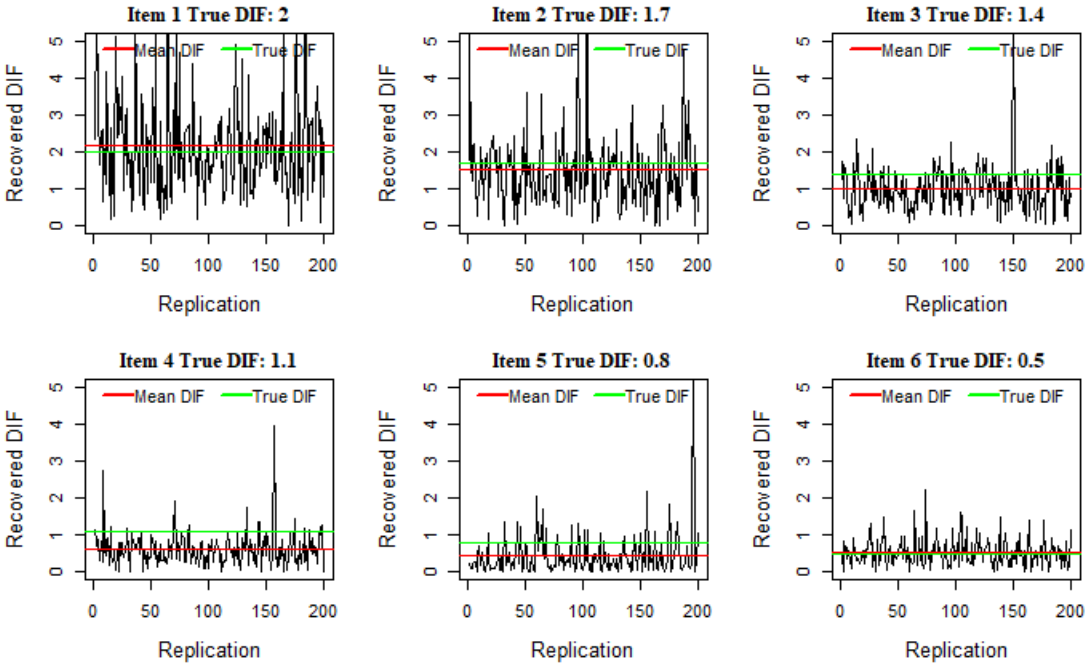

Figure 39b

1006g_lc3_e Item DIF Recovery

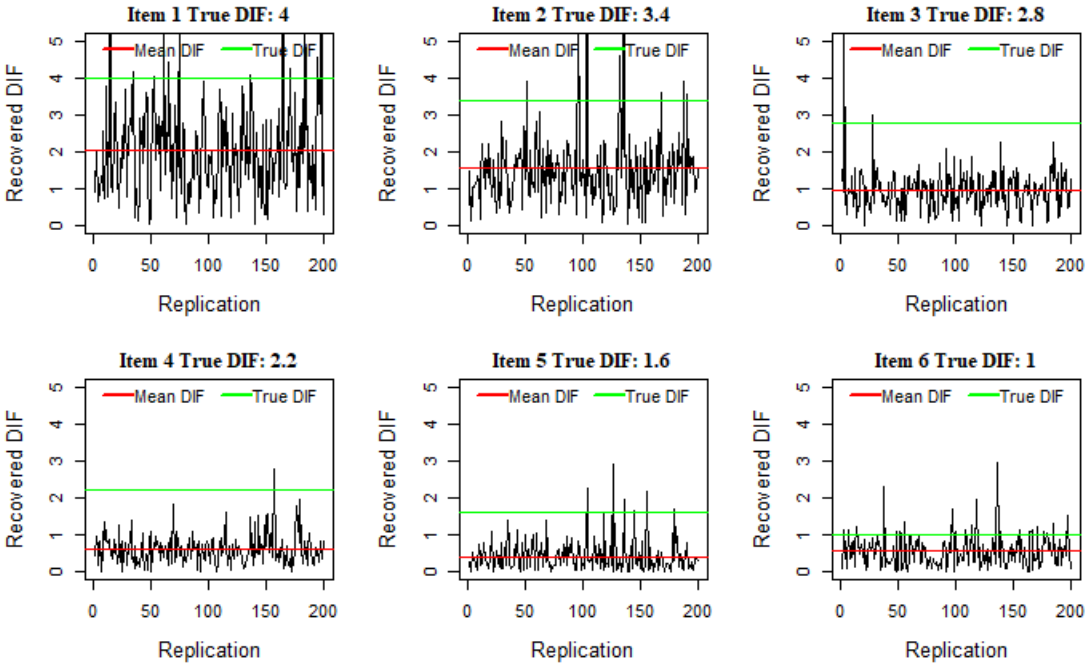

Figure 30a

3006g_lc3_e Item DIF Recovery

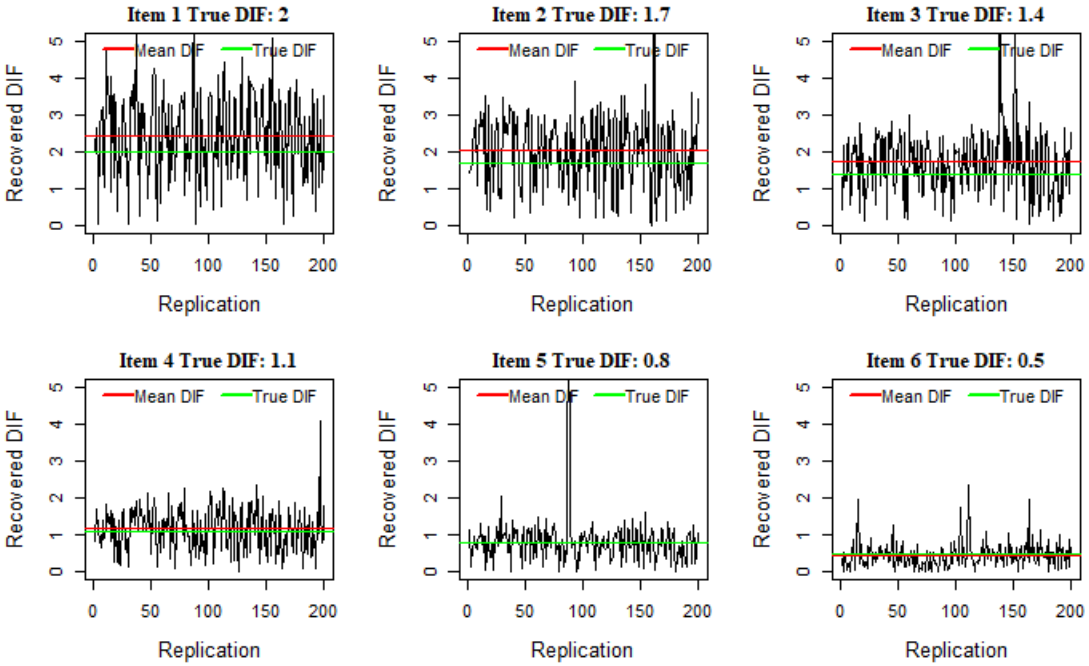

Figure 30b

3006g_lc3_e Item DIF Recovery

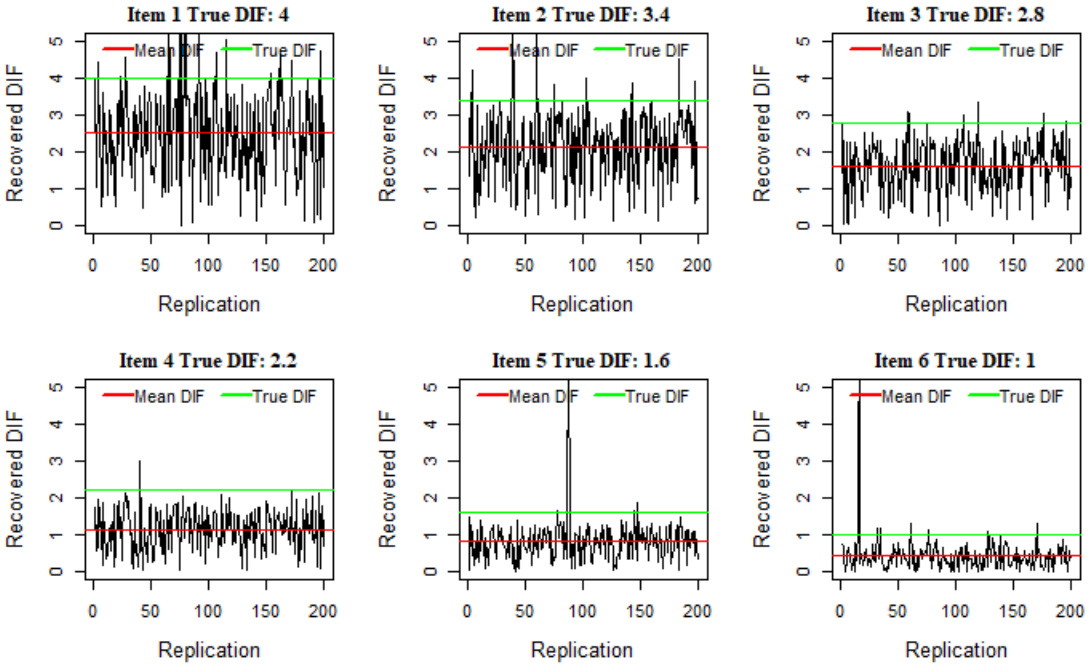

Figure 41a

3012g_lc3_e Item DIF Recovery

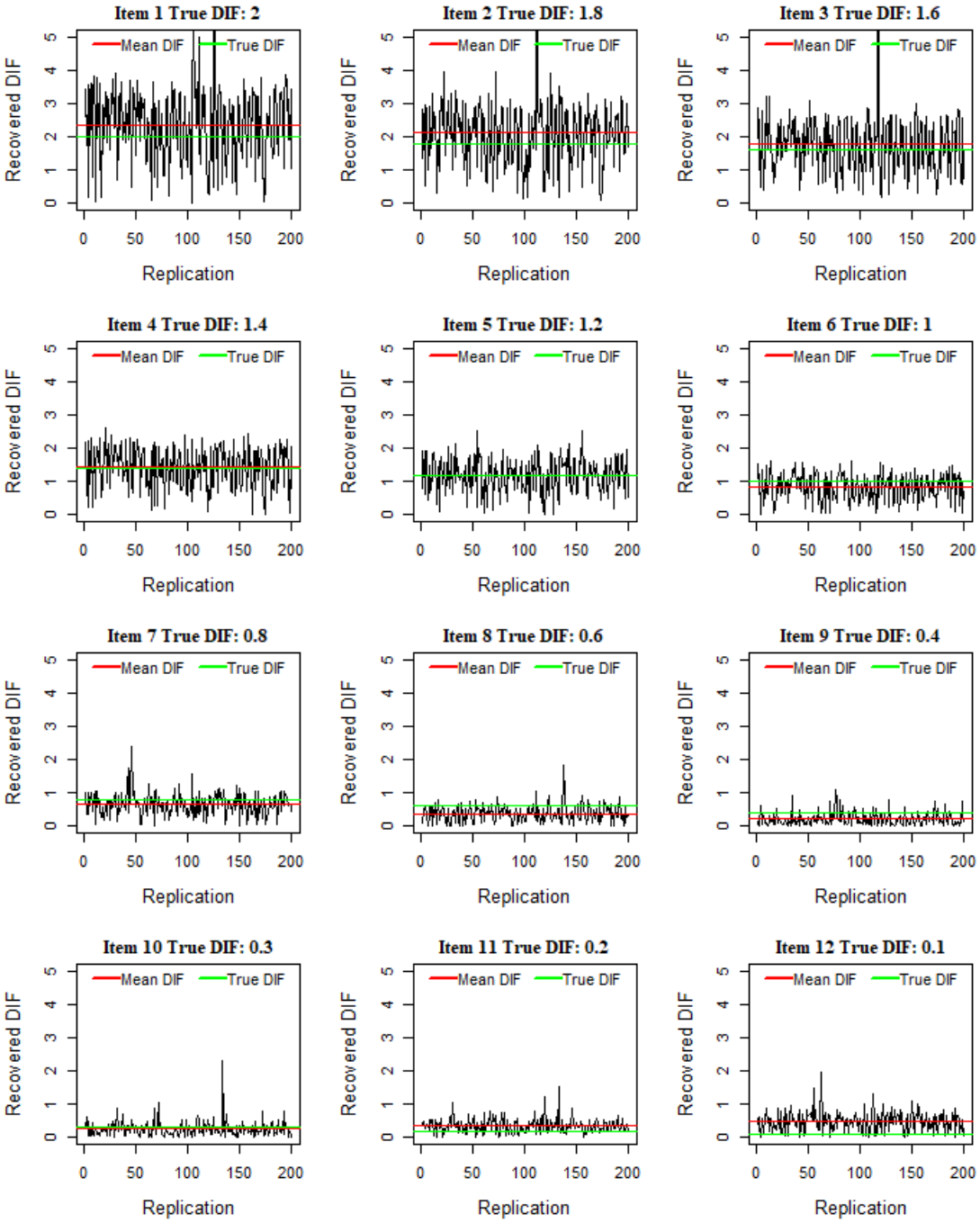

Figure 41b

3012g_lc3_e Item DIF Recovery

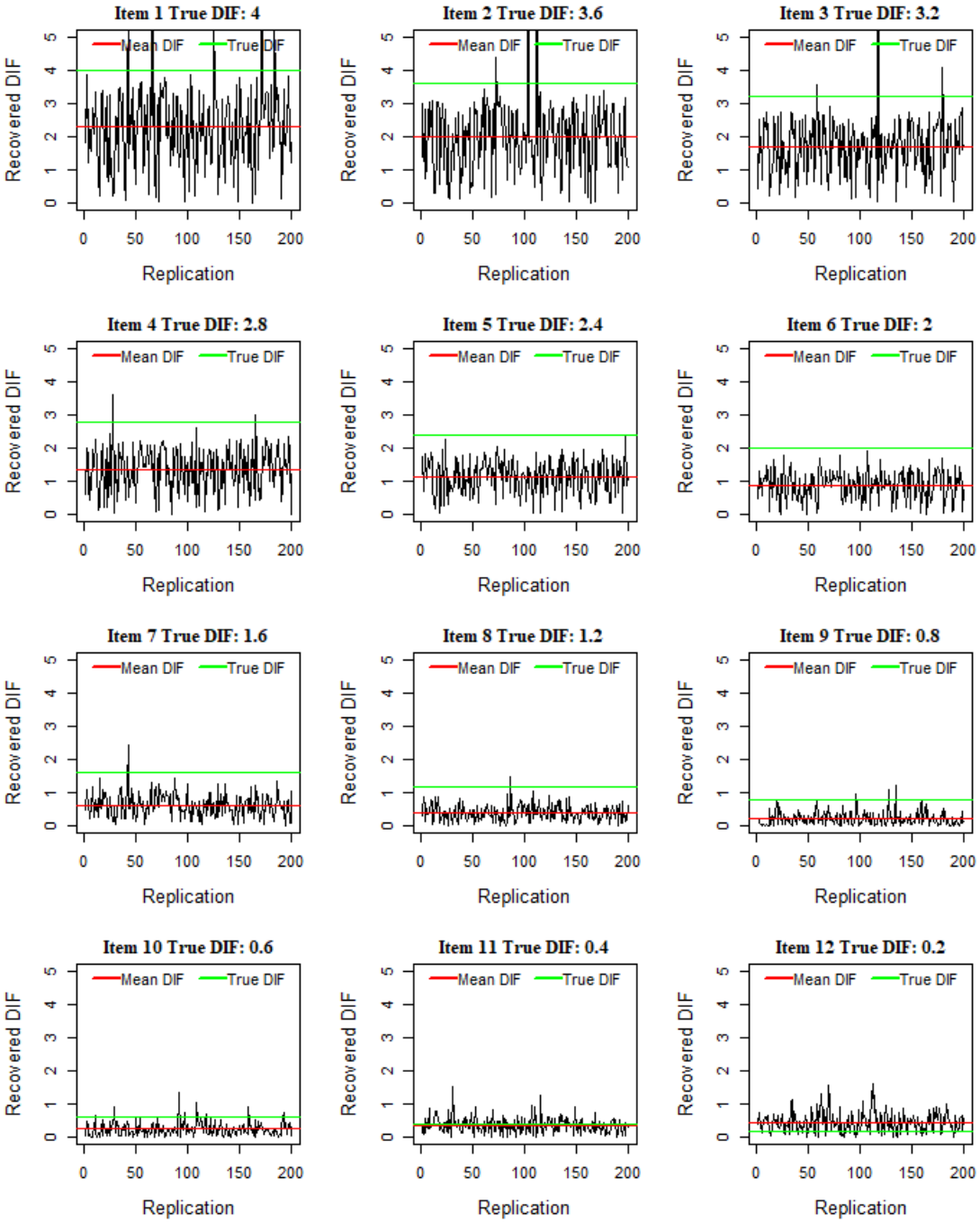

Figure 42a

3018g_lc3_e Item DIF Recovery

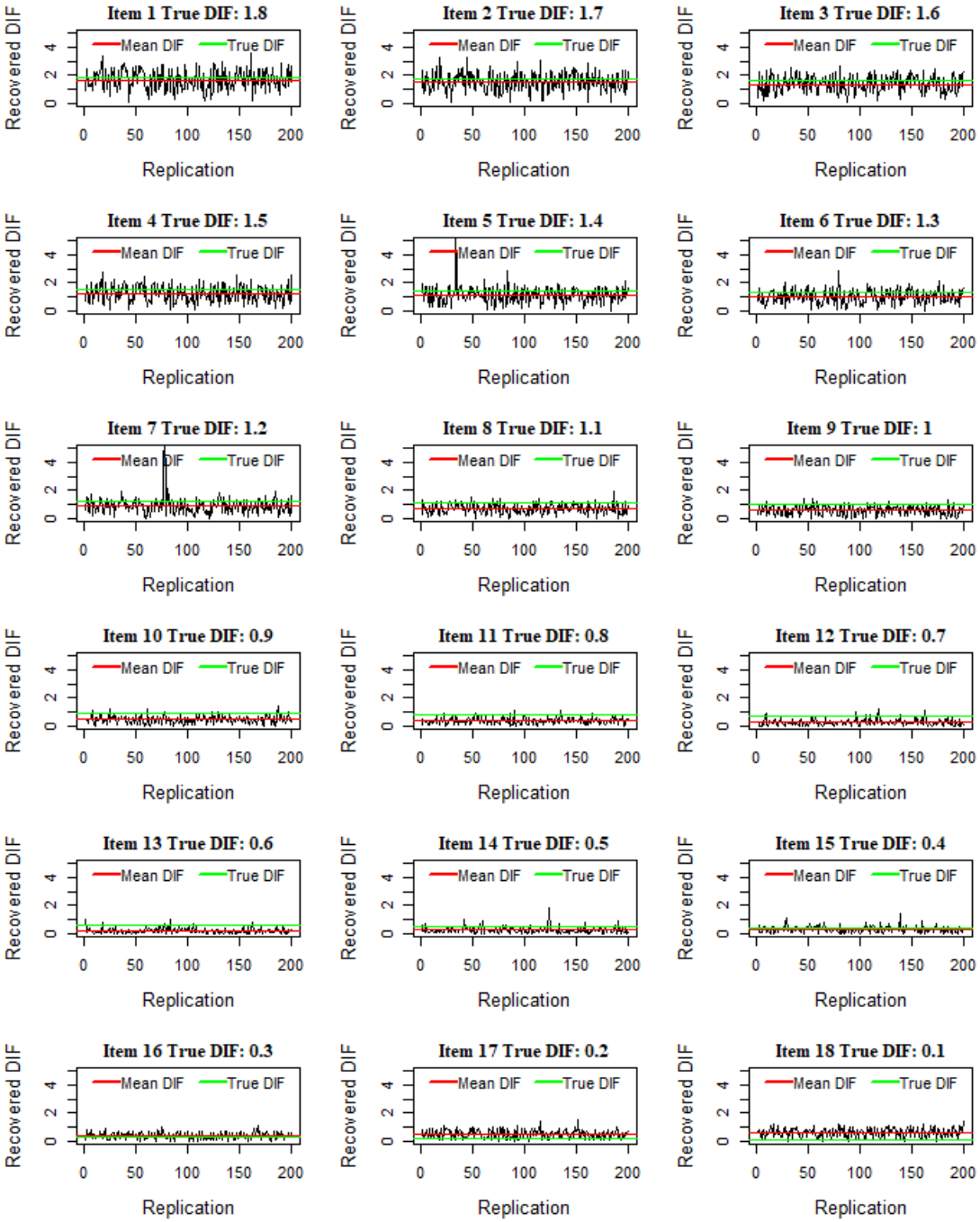

Figure 42b

3018g_lc3_e Item DIF Recovery

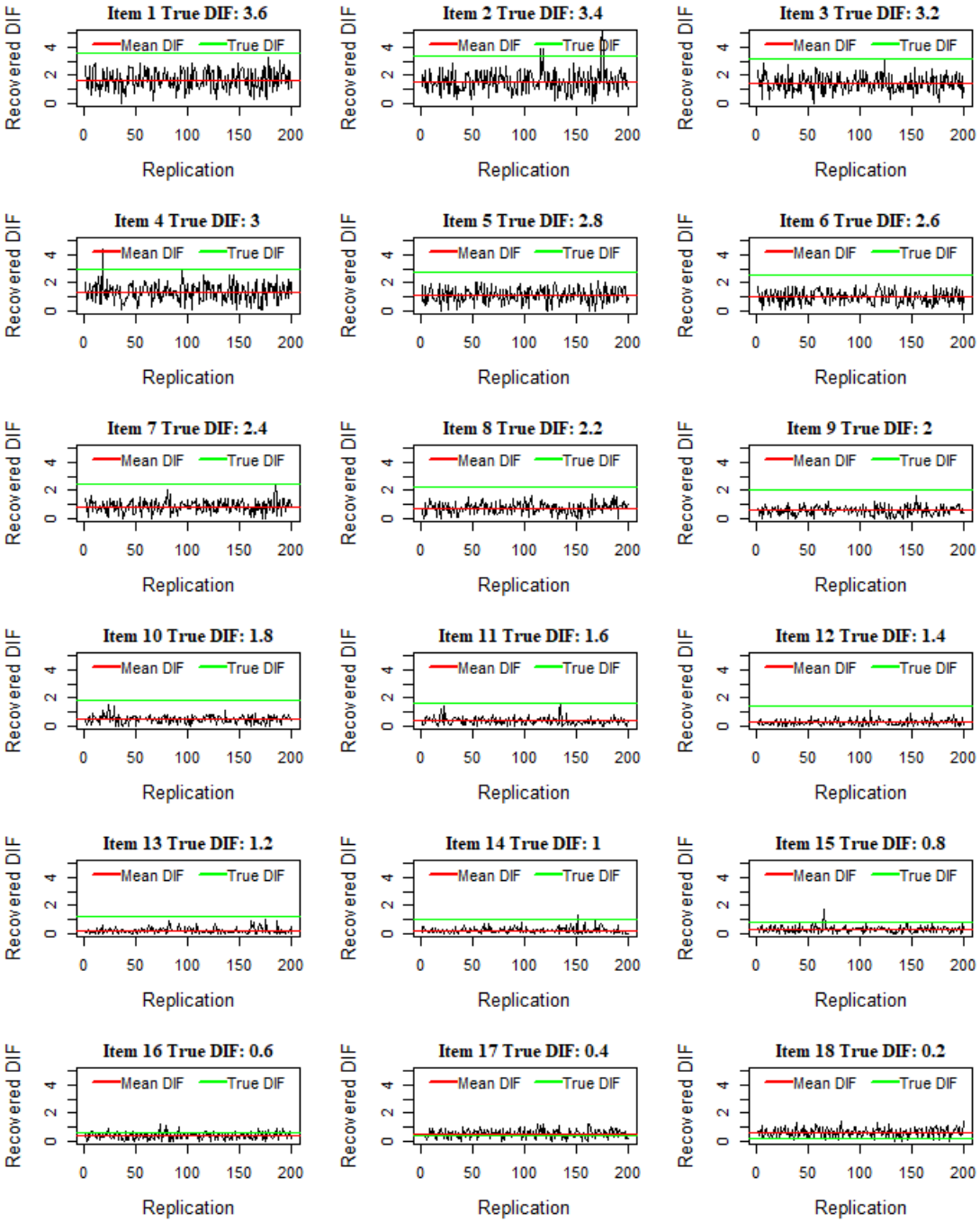

Figure 43a
1002g_lc3_u Item DIF Recovery

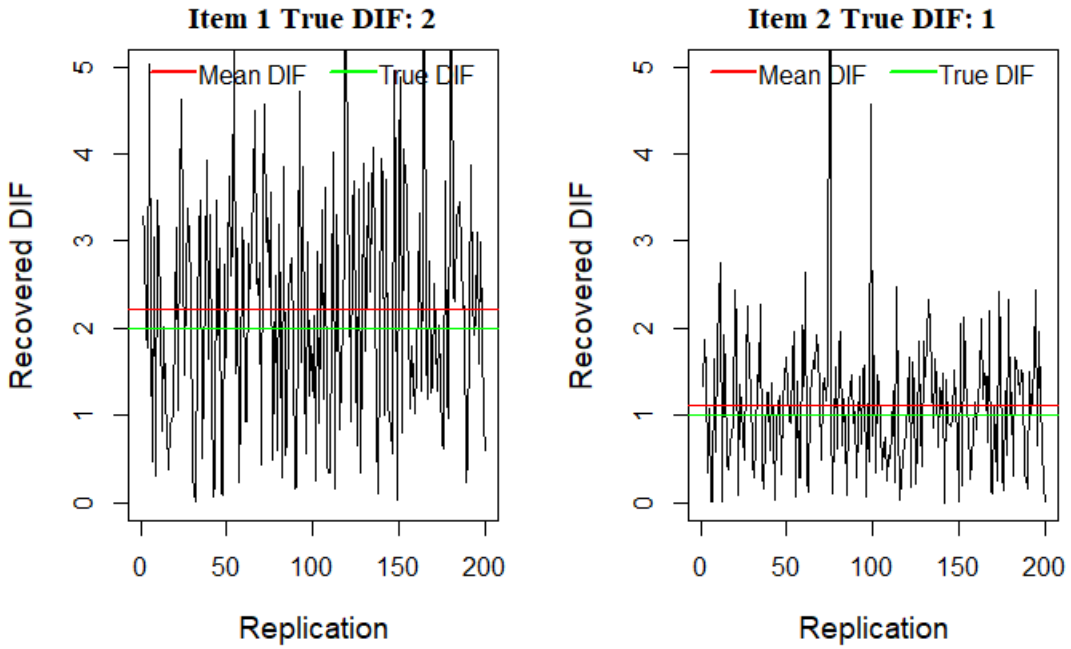

Figure 43b
1002g_lc3_u Item DIF Recovery

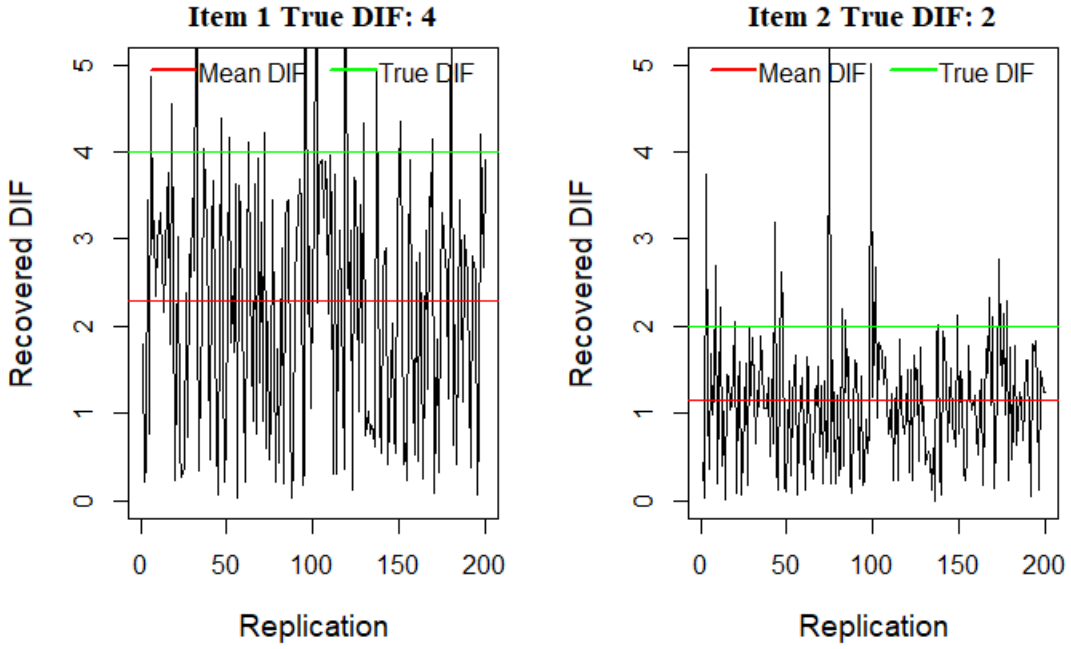

Figure 44a
1004g_lc3_u Item DIF Recovery

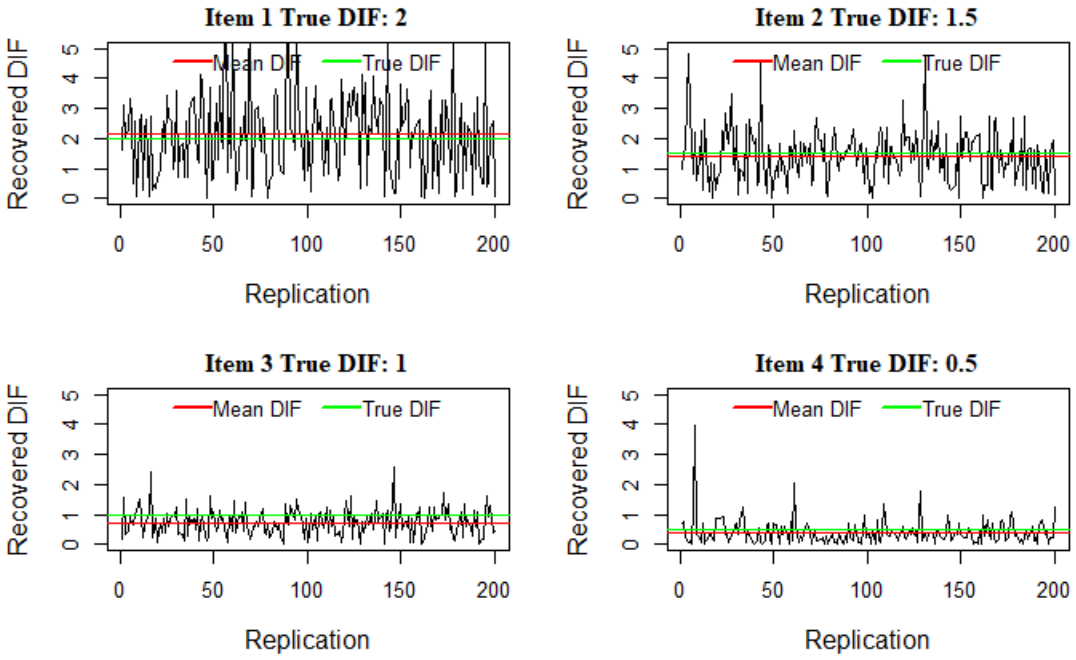

Figure 44b
1004g_lc3_u Item DIF Recovery

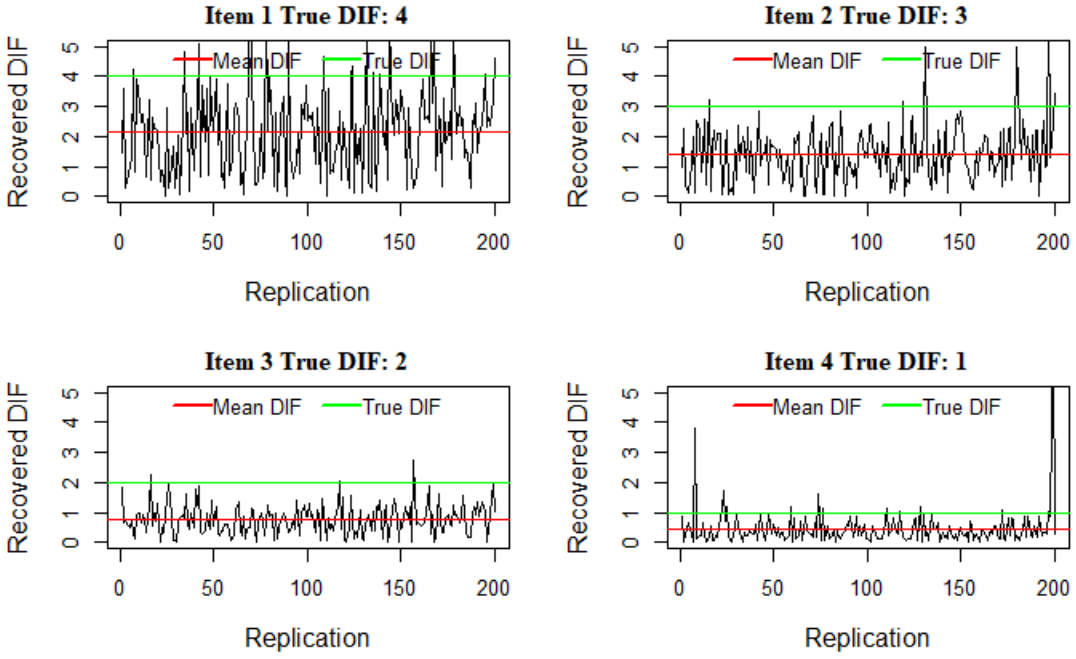

Figure 45a
1006g_lc3_u Item DIF Recovery

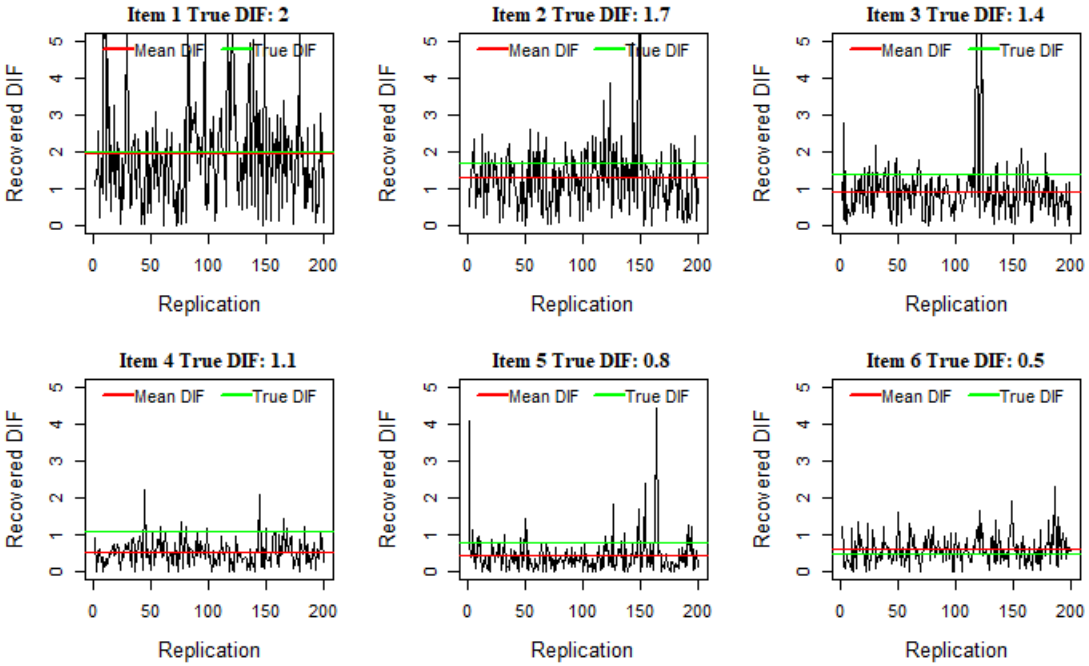

Figure 45b
1006g_lc3_u Item DIF Recovery

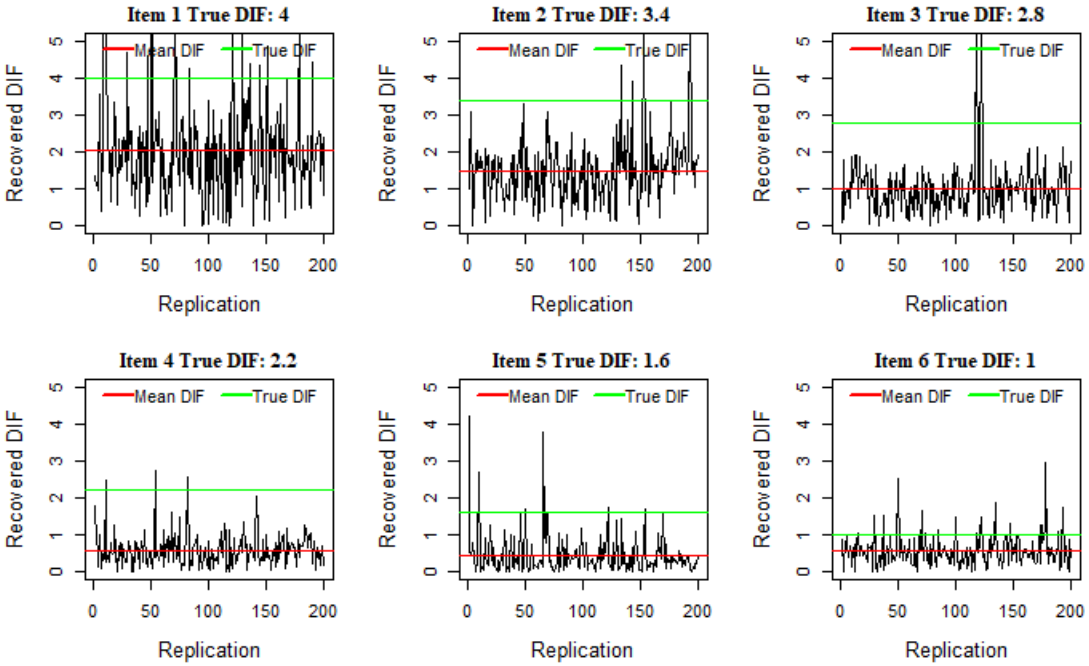

Figure 46a

3006g_lc3_u Item DIF Recovery

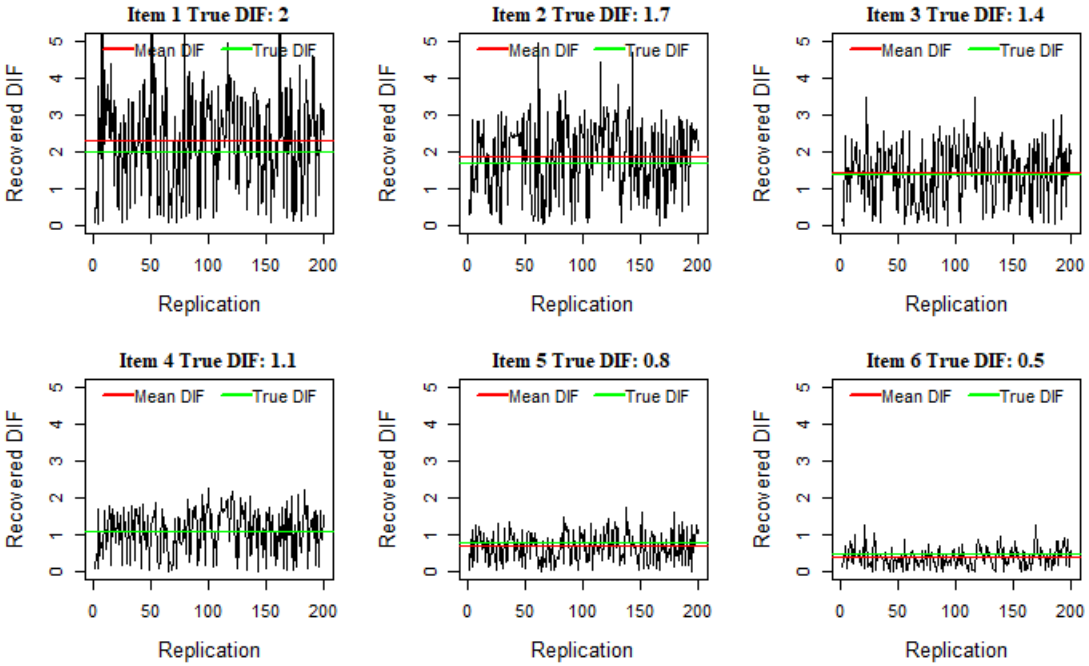

Figure 46b

3006g_lc3_u Item DIF Recovery

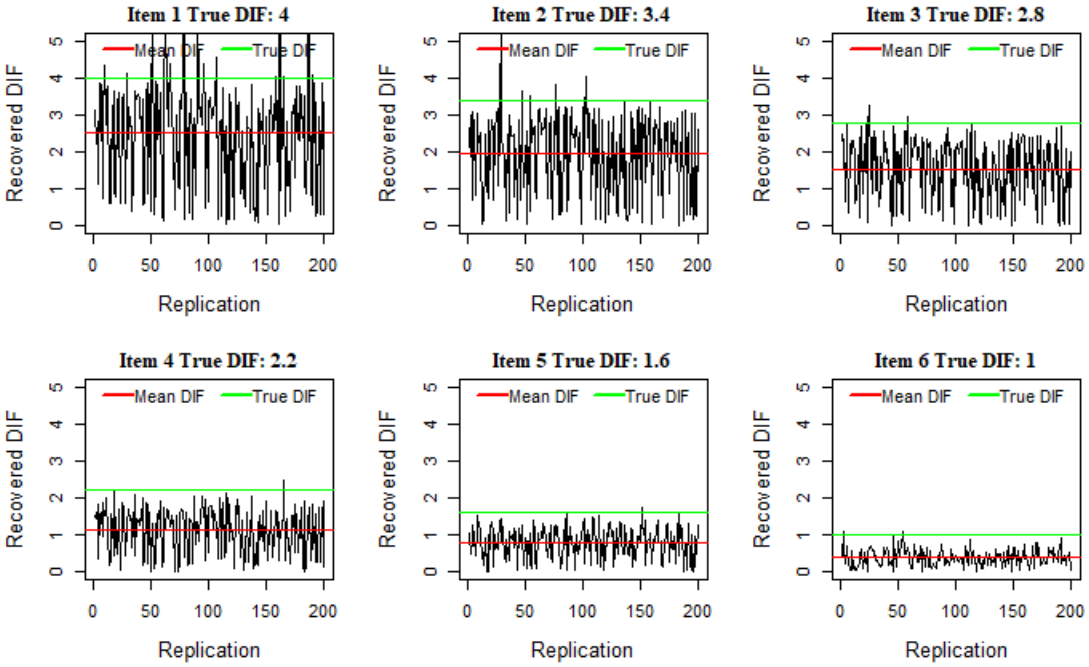

Figure 47a

3012g_lc3_u Item DIF Recovery

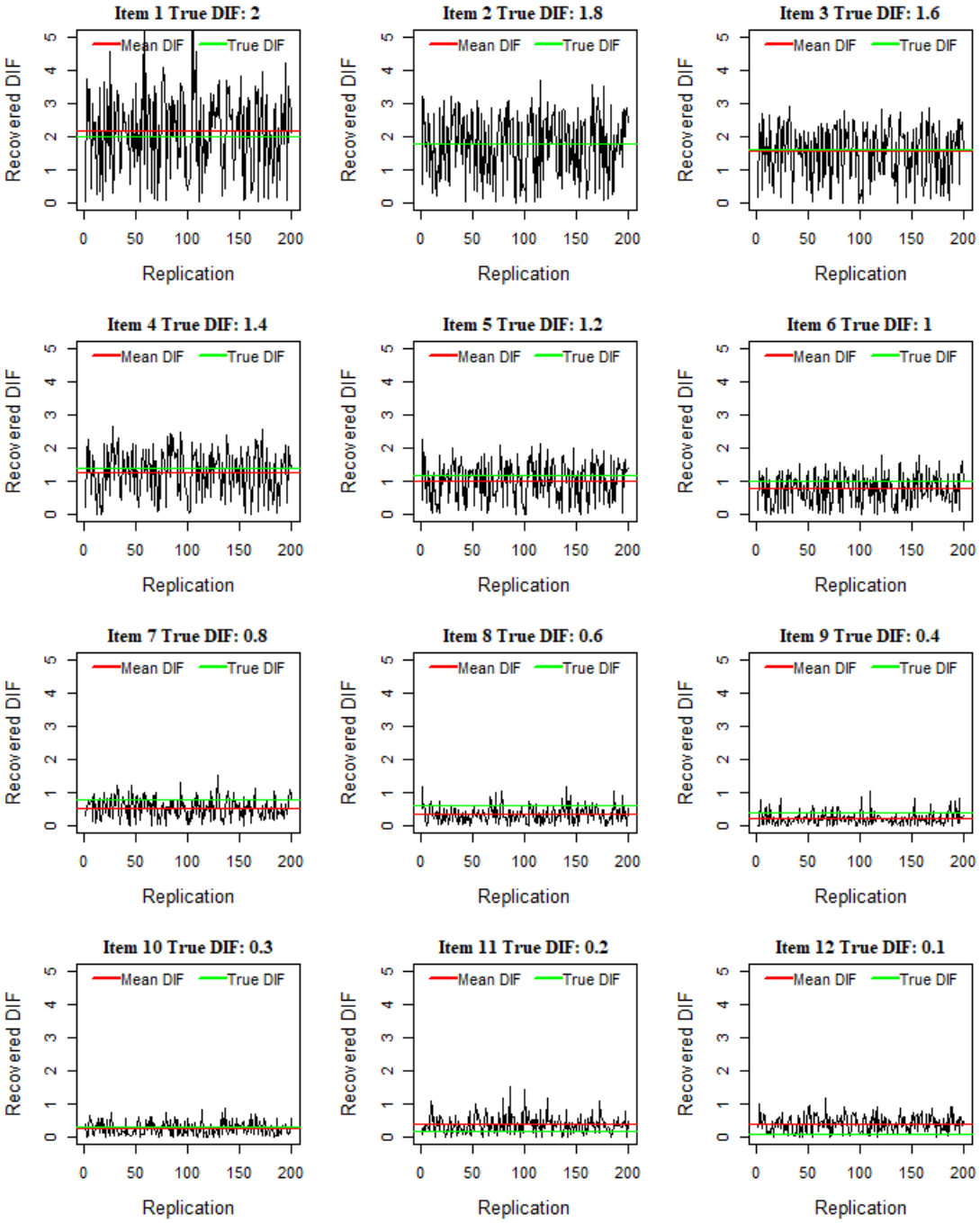

Figure 47b

3012g_lc3_u Item DIF Recovery

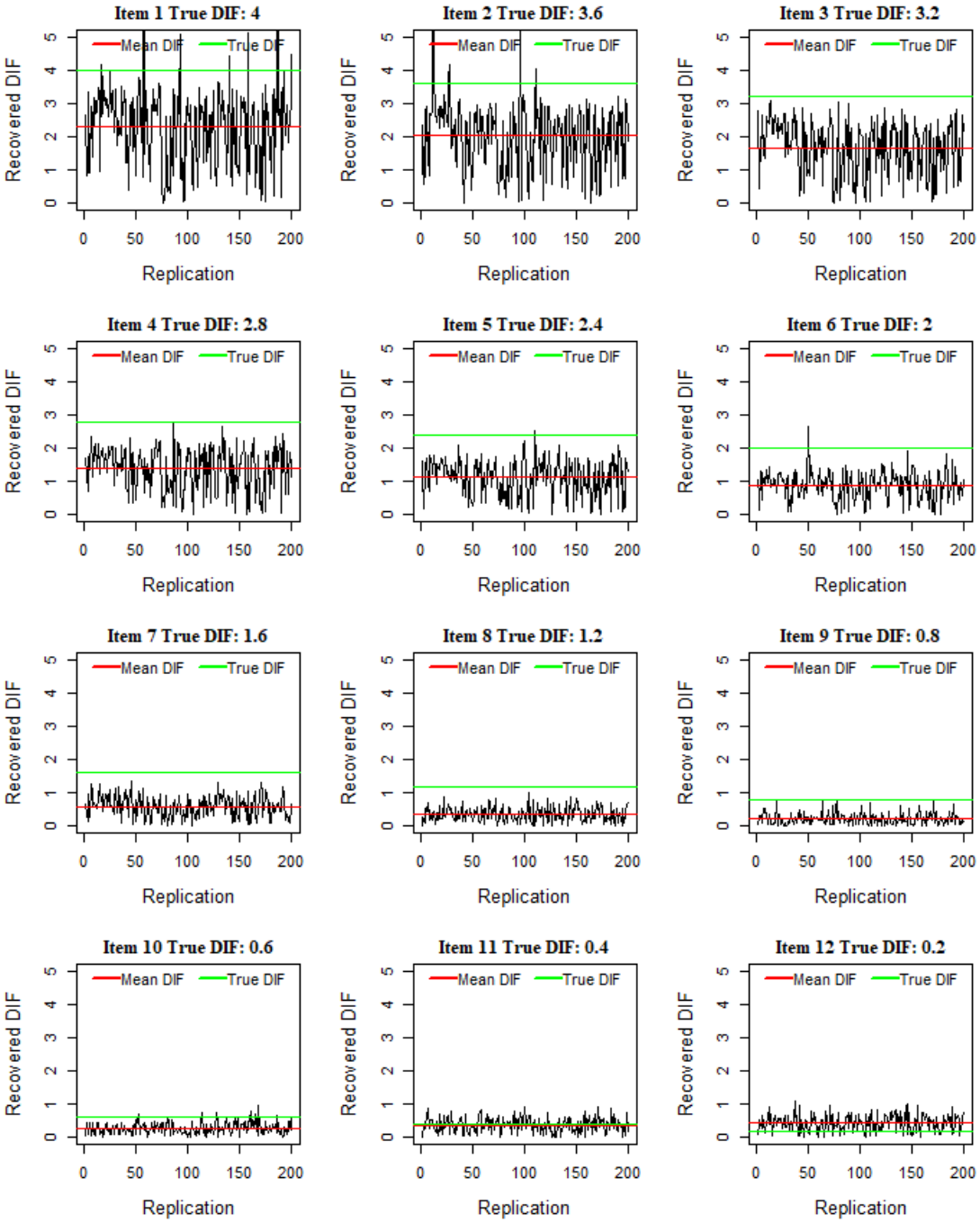

Figure 48a

3018g_lc3_u Item DIF Recovery

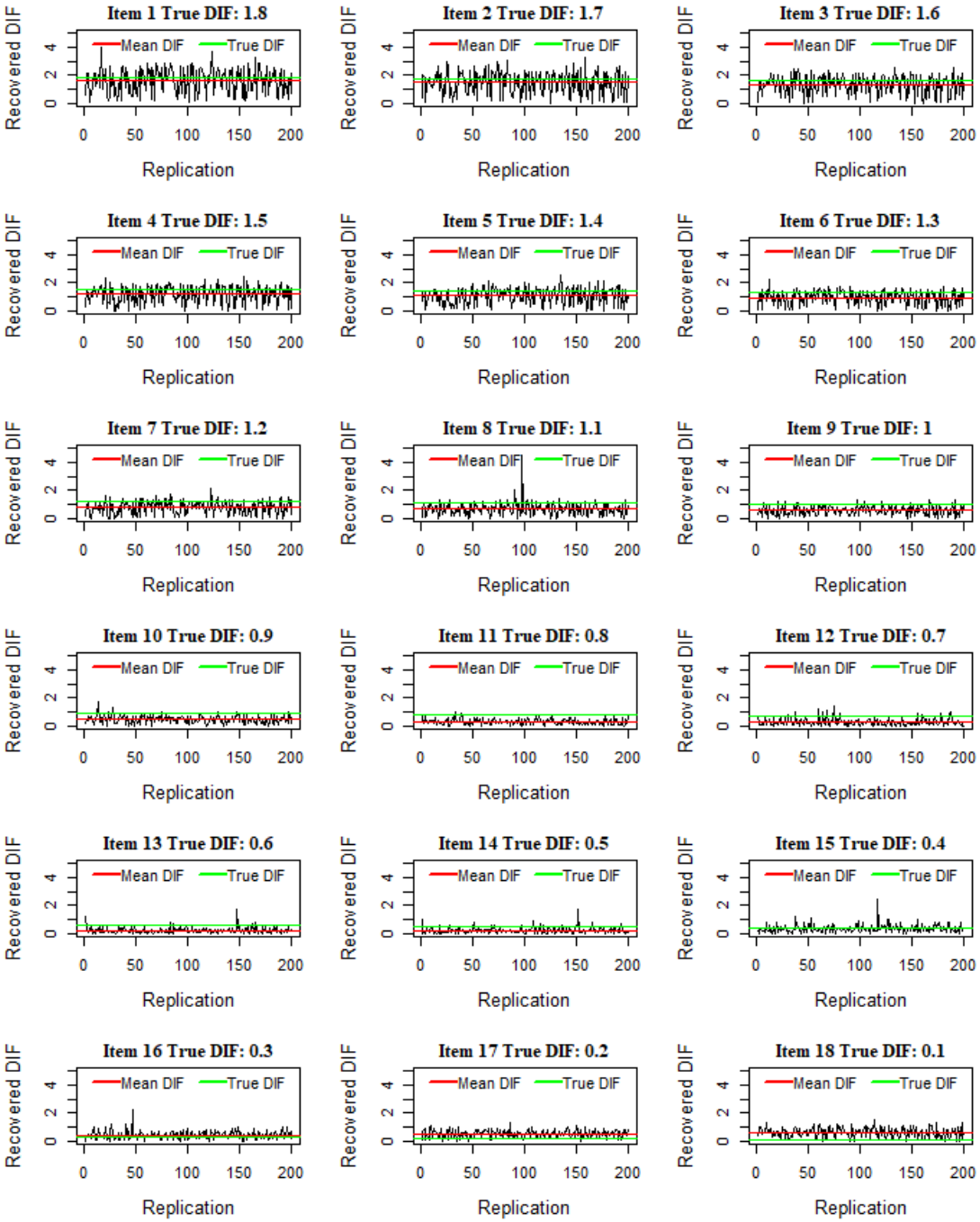

Figure 48b

3018g_lc3_u Item DIF Recovery